Final Year Project Report

Full Unit - Final 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

March 8, 2019

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 Where other published and unpublished source materials have been used, these have been acknowledged.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: 1,587 *2,000* intro + n *3,000* lit review + n *4,000* framework + n *4,000* experiment evaluation + n *1,000* conclusions + n *1,000* professional issues

Student Name: James King

Date of Submission: March 8, 2019

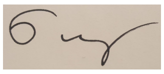
Signature: 

Table of Contents

Abstract

Chapter 1: **Introduction**

Artificial intelligence (AI) has been an idea present in the consciousness of humanity for millennia. From Hephaestus' mighty Talos to Edgar Allan Poe's commentary on 'Maelzel's Chess-Player' the idea has inspired both awe and confusion. As we move away from the mythical and the false, AI embeds itself deeper into our lives and societies. AI techniques are being used for many novel applications in areas such as medicine [24] and game playing [19].

Many of these applications are specifically using agent techniques [49, 22, 7]. Intelligent agents (IAs) have many definitions but one popular definition is that agents are anything that perceives and acts upon its environment [45]. Agents are situated in an environment and many of these agents can be combined in one environment to form a multi-agent system (MAS).

1.1 Motivation

Within MASs it is generally possible for the agents to interact and/or to communicate with each other. Communication generally occurs through agent communication languages (ACLs). The interactions that occur between agents are actions by one or all of the agents in the interaction. There is a possibility that agents can act out of pure altruism and always cooperate with other agents. However, often agents will work to protect their interests i.e. they are selfish.

It is desirable for agents in a MAS to work together to complete tasks and fulfil goals. Therefore, we want to facilitate cooperation between agents which may be selfish. There are analytical tools that we can use in game theory to understand what happens when decision-making individuals interact [33]. Some of these tools are used to explain how cooperation occurs between selfish individuals.

Many of these tools rely on the idea of reciprocal altruism [56], which stipulates that cooperation can occur between selfish individuals if they expect cooperation to be reciprocated. Two such mechanisms are direct and indirect reciprocity. Indirect reciprocity refers to when an agent cooperates with another agent with the expectation that this cooperation will increase the chance of receiving cooperation from others later. Direct reciprocity on the other hand, refers to the expectation of reciprocating cooperation from the agent who initially received the cooperation.

There are many factors to consider surrounding the mechanism. One key factor is the level of visibility of the interactions within the MAS. Nowak and Sigmund [39] limit this visibility to a randomly selected group of individuals in the population (onlookers). Another factor to consider is how information about the interactions can be conveyed. Sommerfeld *et al.* suggested the use of gossip [51].

So how can we use these mechanisms to facilitate cooperation between IAs? What options can we use and how can we make use of them to encourage cooperation between these agents?

1.2 Aims and Objectives

The high level aim of this project is to study how game-theoretic techniques can be leveraged in MASs to create cooperative societies of agents. Derived from this high level aim the objectives of this work are to develop a theoretical framework/model using game-theoretic and MAS techniques inspired by past work in both fields. The next is to implement this framework in a MAS that allows transparent decision making by agents; by which I mean that agents are able to give reasons for their decisions. Another objective is to build this implementation on a distributable platform that allows users to set up games of the model, specify certain variables and view an analysis of the game in order to study how the mechanism and variables affect cooperation in the system.

A feature of the system will be that users are able to create an account and refer back to games they have previously run. I will be using the system and specifically this feature to run my own experiments in order to study the effectiveness of the techniques I have employed and to discover how the system needs to be set up in terms of the variables present. I will then present these experiments and discuss the results with a conclusion on my findings and suggestions of future work as a result of these findings.

Objectives:

1. Develop a theoretical framework
2. Implement this framework
3. Run experiments using the implementation
4. Analyse the experiments and evaluate the results

1.3 Contribution

Much of the work I have reviewed in relation to game-theoretic mechanisms has come from the field of game theory. Because the studies have come from game theorists the implementation of the models devised by the authors have not generally used MAS techniques. My implementation is not only a MAS using a game-theoretic model, but it supports transparent agent decisions due to the use of the logic programming language Prolog for the implementation of agents' decision making components. This implementation is also a web based system, allowing users across the world to experiment with the model I have devised.

Furthermore, the theoretical framework I have designed is inspired by past work on game-theoretic mechanisms, but also builds upon these approaches. One way in which I build upon these is by combining the many aspects of the past approaches such as using MAS techniques to create a game-theoretic model (combining direct and indirect reciprocity) and using gossip as an action to convey reputation information in a mixed reciprocity model.

Lastly, in my system, users associate agents with specific strategies which I have included in the system. These strategies come from past work in the topic of game theory, but are implemented using an agent architecture which I have purpose-designed for my theoretical framework. This arhcitecture included augmenting the original strategies with strategy components for whatever roles agents may have at a particular time-point in a society (such as when an agent is not in an interaction). Another development is the trust models which I have created for the original strategies for interpreting the events in the environment.

1.4 Structure

In the next chapter of this report I have explored and described the past work that has been a central inspiration to the theoretical framework I have devised. I have included other interesting information and possible mechanisms from past work that could be used to encourage cooperation in MASs in the appendix. This background reading then leads into the next chapter in which I have defined the theoretical framework for my system.

Following on from the theoretical framework in the same chapter, I have described the implementation of this theoretical framework in a web application. This application is available for users across the internet to set up games and view the events and analysis of their games. In the next chapter, I use this platform to experiment in order to review the effectiveness of the mechanism, strategies and variables in the game in encouraging cooperation between selfish agents.

The final two main chapters of this report discuss my findings from the experiments and rest of the project. I then conclude and evaluate the project. The appendix is at the end and contains further information referenced in the report but which is not necessarily central to the project.

Chapter 2: **Background**

2.1 Introduction

In the following sections of this chapter I will be exploring and describing past work relevant to this project including MAS techniques and related studies in game theory. First I will describe MAS techniques and concepts that are important to this project including subsections on agents, an agents environment and agent communication. Secondly I will discuss game-theoretical approaches to modelling and studying multi-agent interactions in regards to the evolution of cooperation in MASs.

2.2 Multi-Agent Systems and Intelligent Agents

MASs are a field in computer science concerning the development of societies of agents in which agents are computer systems are computer systems that are capable of deciding on what actions to take in order to reach a delegated goal or the agent's design objectives [57]. These agents typically interact using agent communication languages (ACLs).

2.2.1 Environments

Russell and Norvig [45] state that an agent is designed to work in an environment class or a set of environments. According to Wooldridge [58] an agent is built to fulfil design objectives. These design objectives work within the restriction of the agents environment. Following from this Russell and Norvig [45] define an agent in terms of it's environment, stating that an agent is any system that uses sensors to perceive it's environment and effectors to act upon it's environment.

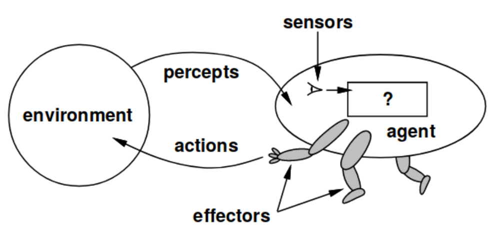


Figure 2.1: Russell and and Norvig's [45] agent

A generic program was designed by Russell and Norvig [45] for how an environment program should run in which there are steps that run in a cycle. In the first step, perception of the environment occurs for each agent in which the agent can use it sensors to receive percepts from the environment. In the next step agents all decide on an action. After all have decided the state of the system is then changed based on the actions that the agents have decided on. A fourth step was added to update the performance scores of each agent.

The effect of these cycle steps is to keep agents synchronised preventing agent actions from changing the state of the environment while other agents are deliberating. A static environment is one that does not change whilst an agent is deliberating. Dynamicity is one of Russell and Norvig's [45] five properties of an environment. The synchronicity of this generic environment program helps keep the environment static, unless other forces (not agent actions) have an effect on the environment's state.

The opposite to a static environment is a dynamic one, in which the environment can change during decision making. Another environment property is accessibility. An accessible environment is one that gives an agent full access to the state of that environment. In an accessible environment there is no need for an agent to keep any internal state about the environment. Unlike in inaccessible environments, in which an agent cannot simply sense all aspects of that environment that are relevant for decision making.

A third property from Russell and Norvig [45] is whether the environment is episodic or not. An episodic environment is split into episodes where the quality of actions in each episode do not depend on previous episodes.

These properties determine how complex an environment is. The final two are whether the environment is deterministic or not and whether the environment is continuous or discrete. If an environment is deterministic it means the effects of an action given some conditions is guaranteed. If an agent can access the whole state of the environment (accessibility) and the environment is deterministic the agent can guarantee the outcome of their action.

An environment is considered discrete actions and percepts are clearly defined and there are a limited number of them. The more properties of an environment that match the following properties, the more complex the environment: continuous, non-deterministic, inaccessible, dynamic, non-episodic.

2.2.2 The Agent Model

Yoav Shoham introduced the agent-oriented programming (AOP) paradigm [48]. This paradigm is similar to object-oriented programming (OOP). In Shoham's formulation of AOP [47] object units are replaced agents, the internal state of a unit is constrained to beliefs, commitments and capabilites, message passing is constrained to using speech act theory to communicate and constraints are placed on methods.

What is an agent?

Since Shoham's formulation there has been much debate as to whether systems are agent systems or simply programs [13]. The exact definition of agency varies depending upon which source is consulted and the application of the concept.

Wooldridge and Jennings [58] gave one of the most influential and powerful descriptions. They put forward a weak notion of agency by laying out four main properties: autonomy, social ability, reactivity and pro-activeness.

Autonomy refers to agents having control over their actions and internal state, and operating without outside intervention. Social ability is the use of an ACL to interact with other entities such as humans and other agents. Responding to environment changes in a timely fashion is what the property of reactivity refers to. Finally pro-activeness is the display of taking initiative to work towards a goal, not simply in reaction to environmental state changes.

Wooldridge and Jennings go on to note that stronger notions of agency endow agents with human-like mental notions. These mental notions represent information in a symbolic model, and make decisions by reasoning with this symbolic model in a symbolic way. This representation and reasoning is no easy feat - known as the representation/reasoning problem [57] - but it does allow agents to make explainable and transparent decisions.

A more up to date and stronger notion of agency may include the use of new developments such as computer vision, natural language processing and deep reinforcement learning concepts. Stronger notions of agency show a tendency towards more humanlike characteristics and AI concepts, drawing clear comparison with a long term goal of AI: to produce systems capable of beating The Imitation Game [27].

Russell and Norvig's Agent Program Components

Russell and Norvig [45] formulated a number of agent program models to describe the internal make up of agents that allow an agent to act in and understand their environment.

The first model is simple reflex agents which keep no internal state, they simply use their percepts from the environment and condition-action rules to decide on actions. These are comparable to reactive agent architectures. Model-based reflex agents use percepts from their environment to form a state which to use alongside condition-action rules and a model of how the state of their world evolved to make decisions.

Goal and utility-based focus on the drives of an agent. The goal based architecture keeps an internal state, and combines this with search and planning decision making components to reach specified goals. States in which an agent is more successful can be formulated in terms of having a higher utility. Utility based agents work to maximise that utility, even in situations when an agent has conflicting goals in which the utility is formed by creating a tradeoff between the conflicting goals.

Learning Agents

Learning agents is the final agent model described by Russell and Norvig. Programs created using this model are not formed by a programmer specifying how to act. Instead these agents use a number of components to feed into a decision making system (performance element). Feedback is provided by the critic component to the learning element which makes changes to the performance element. The problem generator is another component that suggests actions that will hopefully lead to new knowledge.

Machine learning is exploding with new concepts and developments, most notably from the agents systems persepctive deep reinforcement learning. Agents using machine learning techniques often appear to satisfy stronger notions of agency. However, these techniques do not come from a symbolic AI background, and lead to a lack of explainability.

Unifying these techniques with symbolic AI approaches to produce agents which can explain their decisions in order to produce verifiable systems is a keenly researched topic [17, 15]. However, this is an even tougher version of the representation/reasoning problem as we are not only trying to reason with symbolic data, but reason in a complex manner with deeply mathematical techniques.

Deductive Reasoning Agents

Deductive reasoning agents use a form of symbolic AI. The symbolic representation of information is in the form of logical formulae and the reasoning component is done through logical deduction or theorem proving [57].

Shoham's AOP paradigm [48] and Agent0 language [47] were developed to facilitate the creation of deductive reasoning agents. The symbolic representation was formulated in terms of beliefs, commitments and capabilities and agents were given a set of commitment rules to reason about what commitments to make.

Concurrent MetateM is another language for the creation of deductive reasoning agents [12]. The language is known for focusing on creating agents that work by theorem proving. Programmers essentially create deductive theorem provers using temporal logic that dictate agent behaviour.

Deductive reasoning agents suffer from two issues: the transduction problem and the representation/reasoning problem. The transduction problem refers to the difficult in translating from the real world into an accurate and adequate symbolic representation. The representation/reasoning problem are the difficulties in representing symbolic information and reasoning about that information.

Practical Reasoning Agents

Symbolic AI approaches to agency aren't limited to deductive reasoning agents. Practical reasoning approaches focus reasoning towards actions, whereas theoeretical reasoning focuses on beliefs [57]. However this does not mean that practical reasoning models do not incorporate mental notions such as beliefs.

Both deductive reasoning languages such as Agent0 and practical reasoning focused languages and frameworks such as GOAL [20] and PRS (The Procedural Reasoning System) [16] have used the idea of beliefs to base actions on.

PRS uses the belief-desires-intentions (BDI) model [6]. BDI agents keep beliefs in a belief base, these are the information an agent sees as facts about the world and agents are capable of inferring beliefs based on the presence of other beliefs.

Desires are the drives behind an agent's actions. A desire is an objective or state an agent wishes to reach. When an agent is actively pursuing a desire it becomes a goal. These goals must be consistent. When an agent has committed to a desire this desire becomes an intention. Commitment to a desire is when an agent has begun executing a plan towards that desire. These plans are sequences of actions.

Rodriguez *et al.* [44] criticised agent-programming languages (APLs) for forcing the BDI agent architecture upon users of the language, where agent architecture refers to Pattie Maes' description [28]. Their claim is that the use of BDI constrains agents to be endowed with reasoning capabilities that do not always apply to some problem instances. The language SARL - specified by Rodriguez *et al.* [44] - is a more generic language which aims to be architecture independent.

Reactive and Hybrid Architectures

Applications of agent technologies in robotics and embedded systems require architecures and programs to work in resource bound environments. Brooks [8] argues for intelligence as an emergent property of complex systems.

Brooks criticised the use of symbolic AI in the creation of agent systems [8]. He claimed that the abstract reasoning and explicit representations of symbolic AI aren't required for the development of intelligent agents as intelligence is an emergent property of complex systems. In resource bound environments this abstract reasoning and explicit representation are often too resource hungry to be applicable.

In reaction to this Brooks devised the subsumption architecture. In this architecture task-accomplishing behaviours map from percepts to actions, in order to accomplish tasks. These are also known as behviour modules. These modules are able to decide on actions at the same time as each other. Brooks employs a hierarchical structure to select which behaviours decision to use. The lower the layer a behaviour belongs to the greater it's priority.

$$K_1 \rightarrow a_1$$
$$K_2 \rightarrow a_2$$
$$\cdots$$
$$K_m \rightarrow a_m$$

Table 2.1: An example teleo-reactive program

This hierarchical selection of actions is notably similar to teleo-reactive programs [34]. These are programs which contain an ordered set of production rules. These rules map from conditions to actions, and the ordering of them defines which actions are more important.

A criticism of reactive architectures [57] is that through lack of internal state an agent using these architectures can only view the local environment, forcing a short-term view. Hybrid architectures have been developed to combat these issues. One method in which to do this is to use free variables in teleo-reactive program conditions, with which beliefs bind to at run-time.

2.2.3 Multi-Valued Fluent Cached Events Calculus

The aim of the event calculus is to reason about time periods and local events [23]. Events occur at specific timepoints, these events initiate and terminate periods of time for which a fluent holds. For example at time $t1$ event $e1$ occurs, fluent f did not hold prior to $e1$, but $e1$ initiated it and as such it holds after $t1$. A second event $e2$ then occurs at timepoint $t2$ where $t2 > t1$, this event terminates f. Following this event f no longer holds, but f still holds in between $t1$ and $t2$.

The multi-valued fluent cached events calculus (MVFCEC) is an implementation of the events calculus [22]. The implementation improves the speed of querying fluents using a caching system. In Kowalski *et al.*'s [23] original event calculus fluents were considered Boolean. Take fluent F, in the original event calculus $F = true$ if it holds or $F = false$ if not. With multi-valued fluents $F = V$ where V is a value specified by the programmer.

Agents often need to reason about time and states. Many of these agent systems are developed using a logic-based system. The events calculus is a very powerful logic-based tool for this purpose and as such variants of the event calculus have been used in the development of many MASs [2, 22].

The Agent0 belief system reasons about the initiating and terminating of beliefs with certain values through time periods. This belief system is an example of when agents require reasoning about time, and it seems natural to use MVFCEC for the representation of a beliefs system.

2.2.4 Agent Communication

To build cooperative and coordinated societies agent communication is extremely important. In fact it's so important that Wooldridge and Jennings include social ability as one of his four key properties [58]. In the AOP paradigm agents communicate socially via a message passing system.

Many of these languages use speech act theory. Speech act theory uses the idea that communication is an action often to achieve goals and intentions [3]. Both KQML [11] and FIPA ACL use performative verbs to specify the intent of the communication. Both have different performative verbs, but in combination with the message content these languages communicative acts generally fall into one of the five categories of speech acts described by Searle [46].

In human communication we generally have a shared understanding of certain concepts. Such as the definition of a book or plant. In KQML and FIPA ACL shared ontologies are used to define the meaning of commonly used vocabulary of certain subject domains in order to facilitate shared understanding of message content.

The theory of communication is complex. According to Singh [50] a key limitation of existing ACLs (including KQML) is their focus on mental agency over social agency. This is the focus on using the mental state of an agent such as communicating about the beliefs of an agent. Singh claims that this supposes that agents can read each others' mind.

2.2.5 Multi-Agent Interactions

Strategic interaction, rationality, payoff matrices, dominant strategies, nash equilibrium, pareto optimality, social welfare maximisation
Agents are able to interact through communicative actions and by actions in the environment. In Jenning's [21] view of multi-agent interactions, interactions between agents often occur but are not limited to when the agents are linked by organisational relationships. Organisational relationships are subject to change. These agents have their own spheres of influence in an environment, which may overlap each others. An overview of this setup can be seen in figure 2.2.

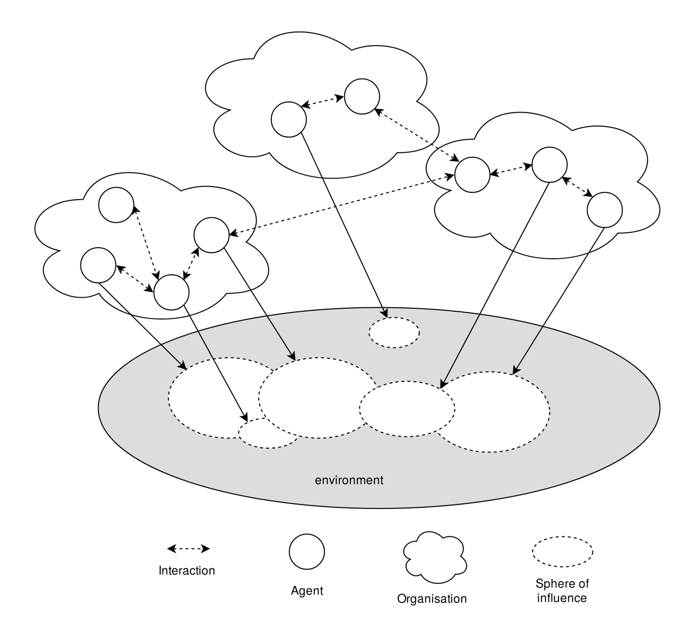


Figure 2.2: Jennings' [21] MAS conceptualisation

When spheres of influences overlap an agent may need to make a decisions that is dependent upon other agents decisions. This overlaps can be described as interactions. Agents in these interactions are assumed to be self-interested, and want to improve their utility. Utility is the the representation of how beneficial events and actions in a system that transform the state of that system are to agents.

When a number of agents choose an action simultaneously the state of the environment is transformed using a state transformer function. Wooldridge [57] formulates agent utility calculations in terms of the outcome of state transformer functions.

Game-theoretic techniques can be used to model and simulate agent interactions using these utility functions. Famously the

2.3 Game Theory

2.3.1 Introduction

2.3.2 Cooperative Phenomena

Game theory formulates mathematical models to study conflict and cooperation between intelligent rational decision-makers [33]. In these mathematical models actions are often defined in terms of cooperation and defection. These models apply to more than just multi-agent interactions, including the natural world.

Early evolutionary theory struggled to explain why cooperation is so prevalent in nature. In fact, it seemed that competition was key to evolution as individuals were competing to survive; the notion famously coined by Herbert Spencer “Survival of the fittest” [52].

Axelrod and Hamilton [4] note two key areas of study that attempt to explain cooperative phenomena in the face of competitions: Kinship Theory and Reciprocal Altruism. These two theories are useful for explaining cooperation in nature, and as such, it may be possible to apply them to MASs in order to facilitate the evolution of cooperation between agents. Due to reasons explained in my appendix in subsection 5.1.1 I will be focusing on reciprocal altruism. edit appendix entry to reflect this

2.3.3 Reciprocal Altruism

Figure 2.3: Cleaning symbioses such as that of the green sea turtle and surgeonfish including the yellow tang show that reciprocal altruism is possible between nonhumans and even interspecies [59]

Reciprocal altruism is an idea most famously put forward by Robert L. Trivers [56]. Trivers defines altruism as behaviour of one organism that benefits another to whom it is not closely related, while being apparently detrimental to the organism performing the behaviour. From this definition and from Trivers’ description we can draw the meaning of reciprocal altruism to be altruism-based on the idea that the altruistic act will be returned.

This idea is a move away from limiting individuals to cooperating with only their kin - a key limitation of Kinship Theory - and towards any individual that they believe their cooperation will be reciprocated by. Axelrod and Hamilton [4] noted this concept as advantageous in explaining cooperation between unrelated individuals, such as is common between humans. I would argue that this concept is also more applicable to higher intelligence societies such as those possible from MASs.

2.3.4 Axelrod, Hamilton and The Iterated Prisoner’s Dilemma

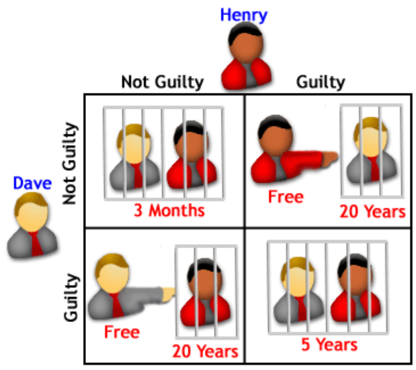


Figure 2.4: A fomulation of the Prisoner's Dilemma payoff matrix [30] similar to Rapoport *et al.* [42]

The prisoner's dilemma was described by Rapoport *et al.* [42] in terms of criminals under interrogation, but Axelrod and Hamilton formulated it in terms of cooperation and defection. This formulation allows the study of a more generic form of interaction betwenn individuals.

In a single round of the prisoner's dilemma two individuals simultaneously choose to either cooperate or defect. After each interactions each individual receives a payoff in the form of a number, the higher the payoff the better. This payoff is dependent upon the actions of both individuals and is illustrated in table 2.2.

From an agents perspective we can view the payoff in terms of utility or what is preferable to an agent [57]. Using player A and player B, with u_i being the utility function for agent i, and the payoff matrix provided by Axelrod and Hamilton [4] we have the following utility functions: $u_A(D,D) = 1$, $u_A(D,C) = 5$, $u_A(C,D) = 0$, $u_A(C,C) = 3$ for player A and $u_B(D,D) = 1$, $u_B(D,C) = 0$, $u_B(C,D) = 5$, $u_B(C,C) = 3$ for player B.

We can order the possible situations by utility for each agent to understand what situations are preferable to each agent. For player A: $(D,C) \geq (C,C) \geq (D,D) \geq (C,D)$. For player B: $(C,D) \geq (C,C) \geq (D,D) \geq (D,C)$. As you can see players are at odds as to which situations they wish to arise. If both attempt to bring about the situation that is preferable to them they will both receive only 1 as payoff each.

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

Table 2.2: The payoff matrix in a typical iterated prisoner's dilemma game (such as Axelrod and Hamilton's [4]). A=x, B=y where x denotes the payoff for A and y denotes the payoff for B.

In the iterated prisoner's dilemma the two agents repeat these rounds and can base their decisions on previous rounds. This allows individuals to use strategies that make use of a mechanism known as direct reciprocity [37], which is a type of reciprocal altruism.

Axelrod and Hamilton [4] used games of the iterated prisoner's dilemma in their round-robin tournaments. These tournaments had a set of players that are associated with strategies. Every player entered into a game against every other player, in which rounds were repeated. For cooperation to be rational in the iterated prisoner's dilemma the 'shadow of the future' needs to present in each round [57], so games were not cut off at a set point but ended by increasing the probability the game won't continue each round.

Axelrod and Hamilton [4] combined these round robin tournaments with a genetic algorithm [31] (genetic algorithms are discussed in subsection 2.3.11). One aim of Axelrod and Hamilton's [4] paper was to review how successful the strategies submitted to them were. The use of a genetic algorithm allowed them to ask three questions of each strategy. Is it robust? Is it stable? Is it initially viable?

Robustness refers to the ability to thrive in an environment with a variety of strategies. Stability refers to the ability to - once fully established - resist invasion by mutant strategies. Initial viability refers to whether or not a strategy can establish itself in a non-cooperative environment.

Axelrod and Hamilton [4] found two strategies with these three abilities: 'tit-for-tat' and 'all defect'. Later on, Nowak and Sigmund [35] found that 'Pavlov' ('win-stay, lose-shift') also has these abilities. The interesting part of 'Pavlov' and tit-for-tat is that they are nice strategies (they begin by cooperating) and that they also actively aid in the evolution of cooperation.

Nowak [37] found that for the evolution of cooperation to occur, the cost-to-benefit ratio of the altruistic act must be less than the probability of another encounter for cooperation to evolve: $w > c/b$. If the probability of subsequent encounters between two individuals is low, then it is very likely that cooperation will not evolve as there will be no sufficient reward. With the advent of huge networks spanning across the world and the drastic increase in devices across these networks, it is highly likely MASs will operate with IAs that are unlikely to re-meet.

This is a definite limitation to the use of direct reciprocity in MASs, and as such I do not see direct reciprocity as a strong contender to encourage cooperation on its own. One required property of a mechanism to encourage cooperation is that it must work when both re-meeting is unlikely and when it is likely. As such, direct reciprocity is not completely inadequate for the problem. However, it is insufficient when not combined with some encouragement for when chances of meeting again are low.

Extensive interest in reciprocal altruism in recent years has been focused towards direct reciprocity, including a number of available libraries such as the Axelrod Python library [10]. I argue that to gain a better understanding of how we can work to facilitate cooperation between IAs, we must look at a wider berth of options.

2.3.5 Indirect Reciprocity

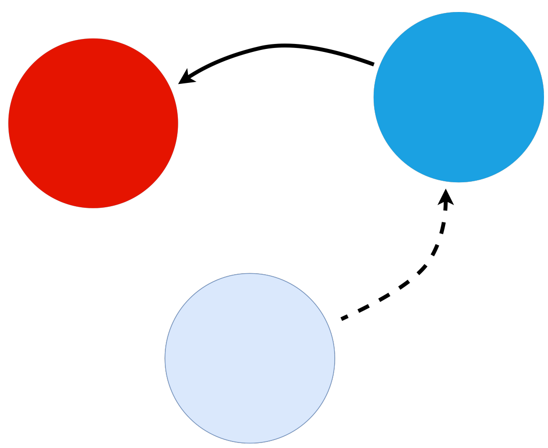


Figure 2.5: The main idea of indirect reciprocity from Nowak [37]

Nowak [37] presents five mechanisms that aim to facilitate the evolution of cooperation. Direct reciprocity is used in the iterated prisoner's dilemma. I discuss three others in my appendix (subsections **??**) and the final one I shall discuss here: indirect reciprocity

This concept is promising for use in MASs as it solves the failure of direct reciprocity's failure with regards to when re-meeting is low. In a large society or a vast network in which agents are likely to operate in success when re-meeting is low is an important property. We will see later on that the two mechanisms can be combined.

Indirect reciprocity uses the group mechanic of reputation to encourage cooperation. Alexander [1], who was an early advocate the idea, focused on human reciprocal altruism, however, subsequent research has abstracted away from the biology [40, 29, 39, 43, 37, 25, 53, 51, 32]. The idea is that if an individual cooperates with another individual, then their reputation will be enhanced in the community. This boost of reputation makes it more likely that they will be helped by others later on. Thus this mechanism a form of reciprocal altruism.

According to Nowak and Sigmund [39] the reputation mechanic requires a higher level of intelligence than direct reciprocity, due to the complexity of group mechanics in the system. It is this kind of higher level intelligence which is required to reason about events in a group that could be a key part in the development of MASs.

Due to this reason, I feel that either indirect reciprocity or possibly a combination of both direct and indirect reciprocity is a good candidate for a mechanism to study further. For the rest of this literature review I will consider both past approaches to indirect reciprocity and other factors in the mechanism that can be used to facilitate the evolution of cooperation.

2.3.6 Nowak and Sigmund

According to Gilbert Roberts [43], Nowak and Sigmund [39] is the most influential model on indirect reciprocity. Therefore, I shall examine this model of indirect reciprocity first. Nowak and Sigmund begin by stipulating that human cooperation is due to people's 'image' of each other, which is comparable to reputation. Nowak and Sigmund converted the image to an integer score between -5 and 5 for simplicity.

The idea is simple, cooperation increases your image score by 1 and defection reduces it by 1. The higher your image score the more likely it is you will receive help. Nowak and Sigmund claim that this mechanism channels cooperation toward valuable members of the society of players.

Nowak and Sigmund note that the use of image scores and indirect reciprocity itself, leave a system open to anticipation, planning, deception and manipulation. These 4 concepts seem closely related to possible happenings in MASs. Deception and manipulation are two factors that I think are worth experimenting with. At the same time, anticipation and planning are a key part of agent design in multiple languages and frameworks. In fact the lack of planning was such an important drawback of the Agent0 language [47] Becky Thomas created a new language - PLACA [54] - which allowed agents to plan, among other improvements.

The framework created by Nowak and Sigmund is simplified from Alexander's [1] idea of human reciprocal altruism. Nowak and Sigmund describe a framework in which there is a population of individuals which act as a pool to select pairs in which one player is the donor - who can choose whether to cooperate or defect - and the other is the recipient of this action. A cooperation costs the donor c to its fitness and benefits the recipient's fitness the value of b where $b > c$. Whereas a defection costs nothing and the recipient is not benefitted. This is shown in the payoff matrix in table 2.3.

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

Table 2.3: The payoff for Nowak and Sigmund's [4] indirect reciprocity model

As noted above, these actions also affect the donor's image score, but Nowak and Sigmund add a caveat when using the idea of onlookers. A specified size group is randomly selected from the population to view an interaction, limiting the spread of reputation information. This concept is displayed graphically in figure 2.6. The concept of onlookers was added due to the realisation of Nowak and Sigmund's that in a group that is spread over a wide geographical area, not all individuals will be able to view each interaction. This sparse nature of interactions is of course especially likely in MASs. Image scores now become one player's view of another rather than a community view of the player. A matrix $ImageScore$ is used to store these scores.

The discriminator is the strategy of choice for Nowak and Sigmund. This strategy stores a number k, and when the individual u using that strategy is a donor to the individual v, u cooperates if the value $ImageScore[u, v] >= k$ otherwise it defects. This strategy is incredibly simple yet effective, and is displayed graphically in figure 2.7. The model also includes the defector and cooperator strategies. Nowak and Sigmund detailed more strategies. These strategies base their decisions not only on that of the recipient's image score but also their own. In this way, the individual will be able to decide whether it is important to boost their reputation in the system in order to receive cooperation or not.

Nowak and Sigmund hypothesize and give evidence supporting the idea that the length of the generation is important to the evolution of cooperation. They highlight that when m donor-recipient pairs are selected in succession in a population of size n, a player is likely to be selected as part of a pair $2m/n$ times. According to their evidence, the higher this value of $2m/n$, the more likely cooperation will evolve.

As previously noted Herbert Spencer coined the phrase "Survival of the fittest" [52]. This principle is put to work by Nowak and Sigmund in their reproduction mechanism. The higher the fitness of an individual the more likely it is that they will reproduce into the next generation. They also include a chance for random mutation in reproduction.

Another idea put forward and evidenced by Nowak and Sigmund is that the evolution of cooperation is dependent upon the donor's knowledge of the image score of the recipient. This chance of knowing an image score is limited by the concept of onlookers. However, in Nowak's 2005 paper on the five rules of cooperation [37] he talks about using 'gossip' as an alternative to direct observation. Gossip and social ability are interesting concepts in MASs. According to Wooldridge and Jennings [58], social ability is a key property in even a weak notion agents.

Nowak and Sigmund stress that discriminators are not 'tit-for-tat' players as used in direct reciprocity, because they use the experience of others. However, the similarity lies in the way that discriminators punish those with lower image scores due to their uncooperative actions, but are forgiving to those who improve and cooperative to those who they consider 'good'. This strategy can also be considered 'nice' if it is using a value of $k <= 1$ as it will cooperate even when the recipient has no past action history.

The model laid out by Nowak and Sigmund is succinct and a good basis for looking at interactions in multi-agent systems. However, there are limitations to their approach in the context of MASs. The first limitation I will highlight is the way in which reputation is implemented. In the version without onlookers each player has a global image score and even in the version with onlookers there is a public matrix encoding all image scores.

Image scores could be seen as a community view of an individual. However, I argue that the idea that image scores are attempting to capture (reputation) is actually more personal. Although there might be a rough consensus as to an individual's general reputation among people in a society, this reputation is often conveyed through social means and is subject to the personal interpretation of each individual.

According to Russell and Norvig [45], IAs use an internal state. In the BDI model [41] the internal state is the beliefs of an agent. It is in this internal state or a belief-system, I argue, that image scores should be stored in order for an agent to have full control over them.

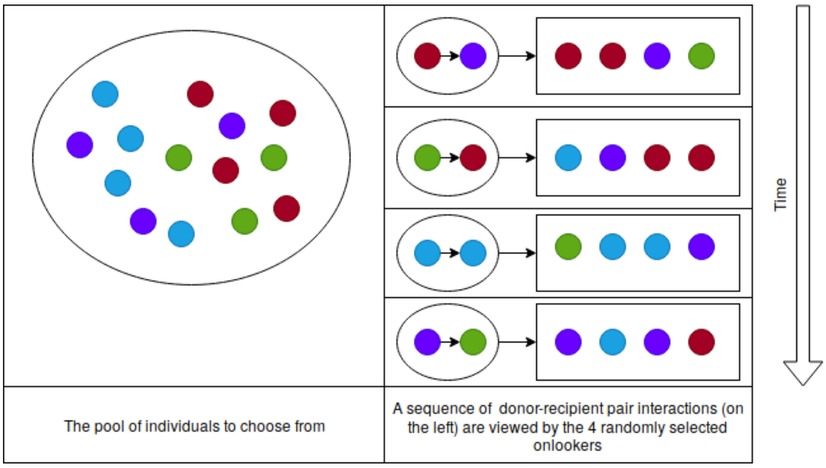


Figure 2.6: The selection of a sequence of donor-recipient pairs and onlookers in Nowak and Sigmund's [39] model of indirect reciprocity. The macro-view of indirect reciprocity. The lines and arrows show the spread of information.

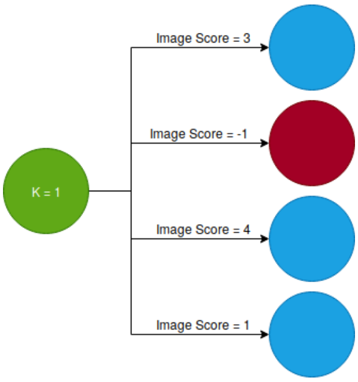


Figure 2.7: On the left an image scoring discriminator with K=1. On the right the agents the discriminator is a donor for. Blue agents represent where there would be a cooperation from the donor, red represents when the discriminator would defect against them. The rule is when $ImageScore >= K$ cooperate else defect.

Multiple trust models could be devised to manage these image scores. This idea is a closer match to the agent paradigm and it would allow for a truly distributed system.

Indirect reciprocity makes a good candidate for use in MASs and this model is both very influential and a good basis for those MASs. The model includes extensive interesting aspects that have been examined carefully including the onlookers and reproduction mechanisms, and also includes ideas that have not been fully researched such as a gossip systems and using deception and manipulation to affect the system. These features allow more room for me to explore if I use Nowak and Sigmund's model as a basis, with a few alterations to more closely match the MAS paradigm.

2.3.7 The Standing Strategy and Further Limitations of Nowak and Sigmund

2.3.8 Mixed Reciprocity Models

2.3.9 Gossip

2.3.10 Mui's Computational Models of Trust and Reputation

2.3.11 Reproduction and Genetic Algorithms

2.3.12 Summary

Chapter 3: **Framework**

3.1 Introduction

In the following sections of this chapter I will be discussing the theoretical side of my framework and the subsequent practical implementation of this framework.

3.2 Theoretical Framework

3.2.1 Introduction

In this section on the theoretical framework I will be presenting an overall design for my mixed reciprocity model and MAS. I have taken inspiration from past work on game theory (discussed in section 2.3) and techniques for the development of MASs (from section 2.2). Franklin and Graesser [13] stipulated that to describe an agent one must describe five aspects: the environment the agent resides in, the agent's sensing capabilities, the possible actions the agent can take, the drives or primative motivators for an agent's actions and the action selection architecture for the agent. I shall describe these five aspects in the following subsections here.

3.2.2 The Game and Environment

The environment in which the agents will reside in my system will provide facilities for interaction between the agents. An instance of the environment in my system will be known as a generation. Each generation contains a set of agents. These sets of agents will act as pools to select donor-recipient pairs and onlookers for those pairs.

A generation is contained within a community. These communities contain multiple generations with a strict ordering. A game has distinct timepoints. The number of these timepoints vary according to the number and length of generations - which are variables set by the user at the start. Say these timepoints range from $1..n$ and there are generations $1..k$ where $k < n$ and $n\%k = 0$. Then the timepoints are distributed evenly across each generation like this:

$$\{\{1, .., (n/k)\},\ \{(n/k+1), .., (2n/k)\},\ ..,\ \{(n-(n/k)+1), .., n\}\}$$

The environment will work using a cycle in which each step follows the process perceive-decide-execute. Each step of the cycle is a timepoint. The perceive section of a step refers to when all agents receive perceptions generated for that timepoint - mostly from actions in the previous timepoint. All agents then decide on an action and the last part of the step is the execution of these actions in the environment.

The key to the cycle steps is that all agents perceive, after which all agents decide and then all actions are executed. The effect is to synchronise agents steps and prevent actions from a timepoint affecting other agents decisions at the same timepoint. Synchronicity keeps the environment static for the period in which an agent is deciding.

In each timepoint there is exactly one donor-recipient pair selected at random from the generation's pool of players. For that pair there is a group of onlookers, again randomly selected from the remaining players in the generation's pool.

As discussed, each generation contains a set of players which participate in the cycle steps for each timepoint of that generation. But how are these sets of players selected? For the first generation, a number of agents and associated strategies (subsection 3.2.7) are selected by the user. For subsequent generations, the set of players is selected using fitness proportionate selection using Lipowski *et al.*'s roulette wheel selection via stochastic acceptance [26] (discussed in subsection 2.3.11).

Each player has a fitness score which is used in the reproduction mechanism. A fitness score starts at zero - it cannot drop below zero - and is affected by the actions of the agent and others as described in subsection 3.2.4. On top of this reproduction algorithm is a chance of mutation c where $0 \leq c \leq 1$ which the user is able to set at the start.

One of the aims of this project is to discover successful agent strategies for the mixed reciprocity model. This problem can be formulated as an optimisation problem, which can be solved using genetic algorithms. The effect of using fitness proportionate selection and the generations of players is to create a genetic algorithm. When experimenting with agent strategies, the strategies which were most successful will hopefully become obvious at the end of a community's generations by the concentration of agents using those strategies.

In summary, the components of this environment are the community, the generations, the sets of agents within the generations, the timepoints throughout the community's life, the onlooker mechanism, the reproduction mechanism, the cycle steps, the donor-recipient pairs, the percepts and the actions. For a discussion on the properties of my environment see appendix subsection 5.2.1.

3.2.3 Percepts

Percepts are the information received at an agent's sensors from it's environment. They are received in the first stage in each cycle step. In my system, percepts are a direct observation of an interaction, hearing gossip from another agent or sensing whether they are the donor or recipient in a donor-recipient pair.

In each timepoint, there is a donor and a recipient selected at random from that generation's pool of players. The two agents are made aware of this fact by receiving a percept of the role that they are taking and which other agent is in that pair in the cycle step's perceive stage. Agents can then act accordingly.

For each interaction, there is a set of onlookers selected at random from the generation's pool of players (not including the recipient or donor). In the cycle step after the interaction takes place the onlookers and the recipient receive percepts containing information as to who the donor and recipient were and the action the donor decided on.

In the gossip (3.2.5) and action (3.2.4) subsections, I discuss an agent's ability to act by gossiping to another agent. The action of gossip produces a percept. This percept contains the message sent using the Simple Agent Gossip Language. The percept is received by the agent given as the recipient by the gossiper.

3.2.4 Actions

My system focuses on interactions between agents and as such, agents require a number of action possibilities by which to interact with one another. To simplify the action there are 3 possible main actions: idle actions, any action when the agent is a donor and gossip actions. Idle actions are simple: an agent is idle in that timepoint, their action has no effect on the environment or other agents, except for through inaction.

Actions when an agent is a donor are more complex. As discussed in subsection 3.2.3, an agent perceives when they are a donor in a donor-recipient pair. When an agent perceives that they are a donor, they have no choice but to commit one of the following two actions: cooperate or defect. In this interaction where the actor is the donor, the recipient has no control over what happens.

The payoff matrix, taken from Nowak and Sigmund [39] in table 2.3, stipulates that when a defect action is chosen, there is no effect on either the donor's or the recipient's fitness. When a cooperate action is chosen however, there is a cost of 1 to the donor's fitness and a benefit of 2 to the recipient's fitness. As discussed in subsection 3.2.3, this action also produces percepts of the interaction event.

When an agent is not a donor, they cannot choose to cooperate or defect with anyone. However, they can choose one of the other two actions: idle or gossip. A gossip action is another type of interaction between agents. A gossiper chooses to communicate with another agent. The contents and structure of this communication are detailed in subsection 3.2.5 and the only effect is the percept discussed in subsection 3.2.3.

3.2.5 Gossip and Simple Agent Gossip Language

```
{
  "recipient": 1,
  "about": 3,
  "gossiper": 4,
  "timepoint": 7,
  "gossip": "positive"
}
```

Source Code 3.1: Example of a message in SAGL

With the design of the gossip in my system, I have focused on social agency over mental agency and keeping the layout of the gossip simple. KQML and FIPA-ACL are powerful tools, which I do not need, and as such I have decided to create my own language which I have named the Simple Agent Gossip Language (SAGL).

I have attempted to create a language which may be used by agents to facilitate three out of four of Baumeister *et al.*'s [5] functions - excluding cultural education, which is beyond the scope of my project. The language I propose is very simple. There are five fields. Three of the fields are identifiers (one for each of the recipient, the target and the gossiper). Another is the timepoint at which the gossip action was decided upon. The final field is the gossip.

Sommerfeld *et al.* [51] categorised gossip in their experiment as either positive or negative. Individuals in my system are attempting to convey similar reputation information, and as such, the gossip field of SAGL either contains the keyword positive or negative.

Agents can attempt to improve bonds between themselves and others, by spreading positive gossip about themselves. Agents can also spread negative information about others to either warn other agents of a non-trustworthy agent, or to harm the agent the gossip is about. Agents can also send positive gossip about others to spread knowledge of trustworthy agents in the system.

3.2.6 Agent Architecture

Russell and Norvig's [45] model-based reflex agent is the closest of their agent architectures to the architecture I will be using. The architecture includes using a trust model to revise beliefs from received percepts and then using these beliefs with teleo-reactive programs to make action decisions.

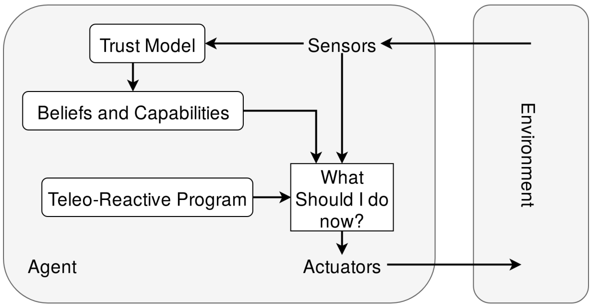


Figure 3.1: The agent architecture in my system.

The trust model and beliefs system use the multi-valued fluent cached events calculus. The beliefs are a combination of the value and the fluent of multi-valued fluents [2]. For example the fluent $standing(agent1, agent2, t) = value$ is the belief of $agent1$ on the standing of $agent2$ at timepoint t. In this fluent the value can either be 'good' or 'bad'. The trust model controls the interpretting of percepts and updating of these fluents.

Capabilities are also a part of the beliefs system that are revised during the interpretation of percepts. These capability beliefs constrain an agent as to the actions to which they can commit i.e. an agent can only commit to an action if they believe that they are capable of it. For example, at a time when an agent is a donor in a donor-recipient pair, they will only believe themself capable of a donor action, constraining their action choice.

To make a decision on an action the agent architecture combines the agent's beliefs with a teleo-reactive program that makes up an agent's strategy. A strategy is made up of strategy 'components'. Of which one I have already discussed: the trust model. The trust model is combined with a component determining how to act as a donor and another determining how to act when not a donor. The decision is a commitment at that timepoint to carry out that action.

3.2.7 Strategies and Trust Models

As I have discussed, an agent's strategy is made up of the trust model, the strategy component for when the agent is a donor and the strategy component for the agent when it is not a donor. These strategies have been inspired by past work on indirect reciprocity and expanded upon. For example, I have enhanced agent's interpretation of events in order that they respond differently when they are the recipient of an action from a donor and when they are the recipient of gossip. The action decisions made when agents are donors have generally remained the same as those in past work. However, these action decision systems have been augmented with new strategy components for when an agent is not a donor.

Defector and Cooperator

Franklin and Graesser [13] talked about an agent's primative motivator or drive. Each strategy has a drive or a combination of drives. For example, the 'Defector' strategy seeks to protect it's own interests by always defecting, thus not causing it to lose fitness points. The defector strategy has been augmented with three possible strategy components which it can use when it is not a donor: 'Lazy', 'Spread Negative' and 'Promote Self'.

The 'Lazy' strategy component always decides on being idle. The 'Spread Negative' component randomly spreads negative gossip about other agents in the system. This component simulates what might be known as 'fake news': a form of deception and manipulation. The final component is 'Promote Self' in which the agent always spreads positive gossip about itself, aiming to increase it's reputation by spreading misinformation and encouraging others to cooperate with the agent using it.

The strategy 'Cooperator' aims to maximise social welfare by always cooperating as a donor. I have also augmented this strategy with three components when the agent agent not acting as a donor: 'Lazy', 'Promote Self' and 'Spread Positive'. 'Lazy' is the same as for the defector strategy and so to is 'Promote Self', although 'Promote Self' when used by a 'Cooperator' could be seen as coming from a drive to improve cooperation and not to deceive other agents. The third is similar to 'Spread Negative' but instead spreads randomly positive gossip in order to improve cooperation between agents in the system. This strategy component could result in accidental manipulation to cooperate with non-reciprocating agents by naive agents.

Image Scoring Discriminator

Neither 'Defector' nor 'Cooperator' have any beliefs about other agents and as such are limited to only trust models that constrain their capabilities. A more insteresting strategy is the 'Image Scoring Discriminator'. Built up around Nowak and Sigmund's [39] discriminator, strategies using this component keep beliefs about the image score of other agents in the system bounded by -5 and 5. To control these image score beliefs I have created three trust models for these discriminators: 'Naive Trusting', 'Trusting' and 'Distrusting'.

The 'Distrusting' trust model abjectly ignores any received percepts which it has not observed directly i.e. rejects any gossip. At the other end of the spectrum, 'Naive Trusting' accepts any gossip it receives, increasing the image score of the agent the gossip is about by 1 if the gossip is positive, and decreasing by 1 if the gossip is negative. The 'Trusting' model is less black and white. This trust model will only change the beliefs regarding the target of the gossip if the gossiper is trusted i.e. has an image score greater than or equal to k, where k is an integer variable set at the start.

The 'Image Scoring Discriminator' strategy component has four corresponding non-donor strategy components. These include 'Lazy' and 'Promote Self'. Again, it could be said that 'Promote Self' is used to improve cooperation towards the agent and not to deceive the recipient.

The other two components are 'Spread Accurate Positive' and 'Spread Accurate Negative'. 'Spread Accurate Positive' spreads positive gossip about trustworthy agents to other trustworthy agents. Trustworthy means that their image score is greater than or equal to k. The aim of this positive gossip is to improve cooperation between trustworthy agents. 'Spread Accurate Negative' spreads negative gossip about agents believed to be untrustorthy - agents with an image score less than k - to trustworthy agents.

It is also possible with agents using image scores to select agents with the 'Personal Grievance' option. When this option is selected, agents using the strategy increase or decrease image scores by 2 when respectively receiving a cooperation or defection. These agents use the selected trust model when they are not the recipient of these actions.

Standing Discriminator

The standing strategy is also included in my system under the name 'Standing Discriminator' with the trust models: 'Naive Trusting', 'Trusting' and 'Distrusting'. This strategy has the following strategy components for when the agent is not a donor: 'Lazy', 'Promote Self', 'Spread Accurate Positive' and 'Spread Accurate Negative'. These are the same as for the 'Image Scoring Discriminator' strategy components except an agent is said to be trustworthy if their standing is 'good'.

The original standing strategy specification for when to change an agent's standing between 'good' and 'bad' is used for directly observed events. The 'Personal Grievance' option is not selectable for the standing strategy.

3.2.8 The Veritability Discerner

The image scoring and standing strategies were both built for systems in which gossip was not present and the reliability of the information they received did not require questioning. I have added trust models to compensate for these issues. However, these trust models are built to either accept or to reject the information from the percept and I do not believe that this procedure reflects how reputation information is interpreted in the real world.

Thus, I have developed a new strategy with corresponding trust models and strategy components. I have named this strategy the 'Veritability Discerner'. An agent using the strategy holds two beliefs about all other agents. One is the number of percepts the agent has received about the other agent n (when the other agent was a donor or the target of gossip). The second, is an integer value I have named the 'veritability rating' v. This rating is initially 0. This rating is similar to an image score but it is not bounded and does not simply increase or decrease by 1, the addition or subtraction to the veritability rating is weighted by the reliability and severity of the percepts.

Trust Models

This strategy comes with three trust models: 'Strong Reactor', 'Balanced Reactor' and 'Forgiving Reactor'. All use the same central weighting and reaction system but apply slightly different weights on top of these.

The central weighting and reaction system is as follows. Viewing of a cooperation is the most reliable indication that the donor can be trusted, and so has the highest weight (20) added to the veritability rating of the agent. Conversely viewing a defection against a recipient who is a trusted agent is the most reliable indication that the donor is untrustworthy, and so the highest weight (20) is subtracted from the veritability rating. When either of these interaction events are observed, the count of percepts received about the donor is incremented by 1. Similaryly to the standing strategy, a defection against an untrusted agent has no effect on the veritability rating of the agent but does increment the percept count.

Negative gossip about an agent from a trusted source subtracts 10 from the veritability rating of that agent, whilst positive gossip from a trusted source adds 10 to the rating. If gossip is from an untrusted source, then only 1 is respectively added or subtracted from the rating.

The 'Strong Reactor' trust model multiplies weights for negative percepts by 2. This multiplication makes the weight for defection against a trusted recipient 40. The multiplication also makes negative gossip from a trusted source subtract 20 from the veritability rating of the target of the gossip. The last effect the multiplication has is to cause a subtraction of 2 from the veritability rating of the target of gossip in response to negative gossip from an untrusted source.

The 'Balanced Reactor' does not apply a weight to either positive or negative percepts. Finally the 'Forgiving Reactor' multiplies positive weights by 2, making the weight of viewing a cooperation 40, positive gossip from a trusted source 20 and positive gossip from an untrusted source 2.

Trustworthiness

A third belief can be derived from the veritability rating and count of percepts received: trustworthiness. An agent using the strategy has a value k similar to an 'Image Scoring Discriminator' agent, though the possible values for this are -10, -5, 0, 5 and 10. The trustworthiness belief is derived as follows. Divide the veritability rating by the count of the percepts received $v/n = t$ - if n is 0 t defaults to 0. If $t \geq k$, then the agent is trustworthy, otherwise they are untrustworthy.

$$
\begin{aligned}
&holds\_at(veritability\_rating(Perceiver, Subject) = V, Timepoint) \land \\
&holds\_at(percept\_count(Perceiver, Subject) = N, Timepoint) \land \\
&get\_K(Perceiver, K) \ \land V/N \geq K \rightarrow \\
&\qquad\qquad trustworthy(Perceiver, Subject, Timepoint)
\end{aligned}
$$

The derivation of this belief has the same effect as taking the mean of a list of weighted actions. This mean takes into account all the actions committed to by an agent but smooths out noise from possibly untrustworthy or distorted sources.

The trustworthiness belief about an agent is used when intepretting gossip and when deciding whether another agent's defection was justified or not. This belief is also used for when the agent is a donor. If the recipient is considered trustworthy then the agent will cooperate, if they are not, the agent will defect.

$$
\begin{aligned}
&get\_strategy(Actor, \text{``}VeritabilityDiscerner\text{''}) \land \\
&interaction\_pair(Actor, Recipient, Timepoint) \land \\
&trustworthy(Actor, Recipient, Timepoint) \ \rightarrow \\
&\qquad\qquad action\_commitment(cooperate, Actor, Recipient, Timepoint)
\end{aligned}
$$

Strategy Components

Finally we need strategy components for when an agent using the strategy is not a donor. These are similar to the discriminator strategies and include: 'Lazy', 'Promote Self', 'Spread Positive Trusted' and 'Spread Negative Untrusted'. 'Lazy' and 'Promote Self' do the same as with all other strategies. 'Spread Positive Trusted' spreads positive gossip to trusted agents about trusted agents. 'Spread Negative Untrusted' spreads negative gossip to trusted agents about untrusted agents.

'Spread Positive Trusted' formalisation:

$$
\begin{aligned}
&get\_strategy\_component(Actor, \text{``}Spread\ Positive\ Trusted\text{''}) \land \\
&not\ interaction\_pair(Actor, \_, Timepoint) \land \\
&findall(Trusted, trustworthy(Actor, Trusted, Timepoint), TrustedAgents) \land
\end{aligned}
$$

$$len(TrustedAgents, Len) \ \wedge Len >= 2 \ \wedge$$
$$get\_2\_random(TrustedAgents, TrustedAgent, Recipient) \rightarrow$$
$$action\_commitment(gossip(\{recipient : Recipient, about : TrustedAgent, gossiper : Actor, timepoint : Timepoint, gossip : positive\}), Actor, Recipient, Timepoint)$$

$$get\_strategy\_component(Actor, \text{“}Spread\ Positive\ Trusted\text{”}) \ \rightarrow$$
$$action\_commitment(idle, Actor, Recipient, Timepoint)$$

'Spread Negative Untrusted' formalisation:

$$get\_strategy\_component(Actor, \text{“}Spread\ Negative\ Untrusted\text{”}) \ \wedge$$
$$not\ interaction\_pair(Actor,, Timepoint) \ \wedge$$
$$findall(Trusted, trustworthy(Actor, Trusted, Timepoint), TrustedAgents) \ \wedge$$
$$len(TrustedAgents, Len) \ \wedge Len >= 2 \ \wedge$$
$$get\_random(TrustedAgents, Recipient) \rightarrow$$
$$action\_commitment(gossip(\{recipient : Recipient, about : TrustedAgent, gossiper : Actor, timepoint : Timepoint, gossip : positive\}), Actor, Recipient, Timepoint)$$

$$get\_strategy\_component(Actor, \text{“}Spread\ Negative\ Untrusted\text{”}) \ \rightarrow$$
$$action\_commitment(idle, Actor, Recipient, Timepoint)$$

Summary

I have presented a new strategy for use in indirect and direct reciprocity models. The aim of this strategy is to be less susceptible to deception and manipulation by weighting the increase and decrease of the ratings by the severity and reliability of the percepts and also by using the smoothing properties of the arithmetic mean.

3.2.9 Summary

In summary, I have presented a MAS that uses a variant of a genetic algorithm to reproduce agents between instances of the environment. I have described the components of the environment and game, the percepts that can be received by agents, the actions available to agents and how these are constrained and the simple agent gossip language. I have also discussed the architecture that I am planning to use in the system and the strategies and trust models available to associate with the agents, including a new strategy that aims to be more suitable with a system in which deception and manipulation are possible.

Bibliography

[1] Richard Alexander. The biology of moral systems (foundations of human behavior). 1987.

[2] Alexander Artikis, Marek Sergot, and Jeremy Pitt. Specifying norm-governed computational societies. *ACM Transactions on Computational Logic (TOCL)*, 10(1):1, 2009.

[3] John Langshaw Austin. *How to do things with words.* Oxford university press, 1975.

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

[5] Roy F Baumeister, Liqing Zhang, and Kathleen D Vohs. Gossip as cultural learning. *Review of general psychology*, 8(2):111, 2004.

[6] Michael Bratman. *Intention, plans, and practical reason*, volume 10. Harvard University Press Cambridge, MA, 1987.

[7] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[8] Rodney A Brooks. Intelligence without representation. *Artificial intelligence*, 47(1-3):139–159, 1991.

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

[10] Vince Knight; Owen Campbell; Marc; eric-s-s; VSN Reddy Janga; James Campbell; Karol M. Langner; T.J. Gaffney; Sourav Singh; Nikoleta; Julie Rymer; Thomas Campbell; Jason Young; MHakem; Geraint Palmer; Kristian Glass; edouardArgenson; Daniel Mancia; Martin Jones; Cameron Davidson-Pilon; alajara; Ranjini Das; Marios Zoulias; Aaron Kratz; Timothy Standen; Paul Slavin; Adam Pohl; Jochen Mller; Georgios Koutsovoulos; Areeb Ahmed. Axelrod: 4.3.0. `http://dx.doi.org/10.5281/zenodo.1405868`, September 2018.

[11] Tim Finin, Richard Fritzson, Don McKay, and Robin McEntire. Kqml as an agent communication language. In *Proceedings of the third international conference on Information and knowledge management*, pages 456–463. ACM, 1994.

[12] Michael Fisher. Concurrent metatema language for modelling reactive systems. In *International Conference on Parallel Architectures and Languages Europe*, pages 185–196. Springer, 1993.

[13] Stan Franklin and Art Graesser. Is it an agent, or just a program?: A taxonomy for autonomous agents. In *International Workshop on Agent Theories, Architectures, and Languages*, pages 21–35. Springer, 1996.

[14] The Photographers from Mpala Live `http://mpalalive.org/sponsors`. Green wood hoopoe. `http://mpalalive.org/field_guide/green_woodhoopoe`, January 2019.

[15] Marta Garnelo, Kai Arulkumaran, and Murray Shanahan. Towards deep symbolic reinforcement learning. *arXiv preprint arXiv:1609.05518*, 2016.

[16] Michael P Georgeff and Amy L Lansky. Reactive reasoning and planning. In *AAAI*, volume 87, pages 677–682, 1987.

[17] Mr. David Gunning. Explainable artificial intelligence. `https://www.darpa.mil/program/explainable-artificial-intelligence`, 2018. Accessed: 8/3/2019.

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

[19] Demis Hassabis and Silver David. Alphago zero: Learning from scratch. *deepMind official website*, 18, 2017.

[20] Koen V Hindriks, Frank S De Boer, Wiebe Van Der Hoek, and John-Jules Ch Meyer. Agent programming with declarative goals. In *International Workshop on Agent Theories, Architectures, and Languages*, pages 228–243. Springer, 2000.

[21] Nicholas R Jennings. On agent-based software engineering. *Artificial intelligence*, 117(2):277–296, 2000.

[22] Özgür Kafali, Alfonso E. Romero, and Kostas Stathis. Agent-oriented activity recognition in the event calculus: an application for diabetic patients. *Computational Intelligence*, 33:899–925, August 2017.

[23] Robert Kowalski and Marek Sergot. A logic-based calculus of events. In *Foundations of knowledge base management*, pages 23–55. Springer, 1989.

[24] Su-Dong Lee, Ji-Hyung Lee, Young-Geun Choi, Hee-Cheon You, Ja-Heon Kang, and Chi-Hyuck Jun. Machine learning models based on the dimensionality reduction of standard automated perimetry data for glaucoma diagnosis. *Artificial Intelligence in Medicine*, 18, 2019.

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

[26] Adam Lipowski and Dorota Lipowska. Roulette-wheel selection via stochastic acceptance. *Physica A: Statistical Mechanics and its Applications*, 391(6):2193–2196, 2012.

[27] Computing Machinery. Computing machinery and intelligence-am turing. *Mind*, 59(236):433, 1950.

[28] Pattie Maes. The agent network architecture (ana). *SIGART Bull.*, 2(4):115–120, July 1991.

[29] Theo C. M. Bakker Manfred Milinski, Dirk Semmann and Hans-Jürgen Krambeck. Cooperation through indirect reciprocity: image scoring or standing strategy? *Proceedings of The Royal Society*, 268:2495–2501, May 2001.

[30] Arjun Mani. What is the prisoner's dilemma in layman's terms? what are its applications? what are it's applications? `https://www.quora.com/What-is-the-prisoners-dilemma-in-laymans-terms-What-are-its-applications`, September 2013.

[31] Melanie Mitchell. *An introduction to genetic algorithms.* MIT press, 1998.

[32] Lik Mui. *Computational models of trust and reputation: Agents, evolutionary games, and social networks.* PhD thesis, Massachusetts Institute of Technology, 2002.

[33] Roger B Myerson. *Game theory.* Harvard university press, 2013.

[34] Nils Nilsson. Teleo-reactive programs for agent control. *Journal of artificial intelligence research*, 1:139–158, 1993.

[35] Martin Nowak and Karl Sigmund. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. *Nature*, 364:56–58, July 1993.

[36] Martin A Nowak. *Evolutionary dynamics.* Harvard University Press, 2006.

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

[38] Martin A. Nowak and Robert M. May. Evolutionary games and spatial chaos. *Nature*, 359:826–829, 1992.

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

[40] Steve Phelps. An empirical game-theoretic analysis of the dynamics of cooperation in small groups. *Journal of Artificial Societies and Social Simulation*, 2016.

[41] Anand S Rao, Michael P Georgeff, et al. Bdi agents: from theory to practice. In *ICMAS*, volume 95, pages 312–319, 1995.

[42] Anatol Rapoport, Albert M Chammah, and Carol J Orwant. *Prisoner's dilemma: A study in conflict and cooperation*, volume 165. University of Michigan press, 1965.

[43] Gilbert Roberts. Evolution of direct and indirect reciprocity. *Proceedings of The Royal Society*, 275:173–179, September 2008.

[44] Sebastian Rodriguez, Nicolas Gaud, and Stéphane Galland. Sarl: a general-purpose agent-oriented programming language. In *2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, volume 3, pages 103–110. IEEE, 2014.

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

[46] John R Searle and John Rogers Searle. *Speech acts: An essay in the philosophy of language*, volume 626. Cambridge university press, 1969.

[47] Yoav Shoham. Agent0: A simple agent language and its interpreter. In *AAAI*, volume 91, page 704, 1991.

[48] Yoav Shoham. Agent-oriented programming. *Artificial intelligence*, 60(1):51–92, 1993.

[49] Barry G Silverman, Nancy Hanrahan, Gnana Bharathy, Kim Gordon, and Dan Johnson. A systems approach to healthcare: agent-based modeling, community mental health, and population well-being. *Artificial intelligence in medicine*, 63(2):61–71, 2015.

[50] Munindar P Singh. Agent communication languages: Rethinking the principles. *Computer*, 31(12):40–47, 1998.

[51] 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.

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

[53] Robert Sugden et al. *The economics of rights, co-operation and welfare*. Springer.

[54] Sarah Rebecca Thomas. Placa, an agent oriented programming language. 1993.

[55] Arne Traulsen and Martin A. Nowak. Evolution of cooperation by multilevel selection. *Proceedings of the National Academy of Sciences of the United States of America*, 103:10952–10955, 2006.

[56] Robert L Trivers. The evolution of reciprocal altruism. *The Quarterly review of biology*, 46(1):35–57, 1971.

[57] Michael Wooldridge. *An introduction to multiagent systems.* John Wiley & Sons, 2009.

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

[59] Sea Turtle World. Green sea turtle cleaning station. https://www.seaturtle-world.com/green_sea_turtle_cleaning_station/green_sea_turtle_cleaning_station_600/.

Chapter 4: **Professional Issues**

are current machine learning methods unethical? Deep symoblic reinforcement learning you can query actions, like I have done.

Chapter 5: **Appendix**

Describe the contents of my appendix.

5.1 Background

5.1.1 Kinship Theory

Axelrod and Hamilton [4] described the way in which cooperation in nature (with the exception of homo-sapiens) is almost always between related individuals. An earlier paper by Hamilton [18] argues that individuals don't only work toward improving their own fitness, but towards what Hamilton defines as 'inclusive fitness'. Inclusive fitness is the sum of a player's fitness and the fitness of each of their relations multiplied by a coefficient. The coefficient used by Hamilton is Wright's coefficient of relatedness, as illustrated in figure 5.1. It could be possible to create a similar coefficient of relatedness for use in a MAS.

Richard Dawkins [9] advocated for the idea of the selfish gene. From a biological perspective, this idea postulates that actors are hardwired to propagate their genes. Dawkins asserts that this drive is due to the fact that genes are the true replicators evolutionarily rather than the actors themselves. Those with a high coefficient of relatedness to an individual are far more likely to carry their genes and to help them proliferate. This mechanism is similar to that presented by Hamilton [18], but with a biological backing. From a biological perspective this may make sense. However, it does not seem natural to translate an agent's strategy to the idea of genes.

Further, although it is possible to create a coefficient and an idea of relatedness similar to that of Hamilton's model [18] for a MAS, it does not seem a natural translation. Another limitation to the use of kinship theory for MASs is that systems are ideally inclusive of individuals that can contribute to the society. For example, if an agent is looking to actively contribute to a society, but is not kin with the members, a MAS using kinship theory would exclude them and thus limit the abilities of that society.

Furthermore, Axelrod and Hamilton [4] highlight that humanity is the exceptional society which does not limit itself to cooperating mostly only with kin. I would surmise that this exception is due to the higher level of intellect of homo-sapiens in comparison to other species. Many have suggested that the capabilities of AI could match or even surpass the intelligence of humans. Therefore, I would suggest that societies of IAs should also not be limited to the use of kinship theory to facilitate cooperation.

As such, I shall not be using kinship theory for my theoretical framework. I will be aiming to use a mechanism that is inclusive of agents that aim to become valuable members of the society and also a mechanism which fits naturally into the agent's paradigm.

5.1.2 Network Reciprocity

Nowak - in his paper 'The Five Rules of Cooperation' [37] - identified and compared five key mechanisms that can aid in the evolution of cooperation, two of which I have already discussed (direct reciprocity in section 2.3.4 and kin selection in **??**). The other three are

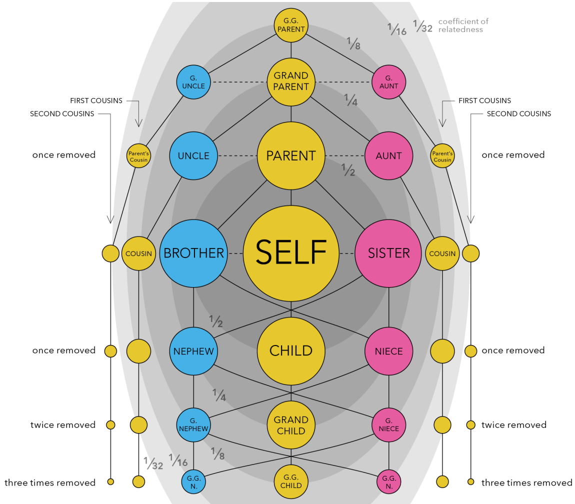


Figure 5.1: Wright's coeffcient of relatedness by Citynoise - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=37723128

Figure 5.2: The Green Wood-Hoopoe native to Africa participates in cooperative breeding as the bird not only looks after its own chicks, but those of other breeding pairs [14]

network reciprocity (which I shall examine here), group selection and indirect reciprocity (both of which have their own sections).

Network reciprocity uses a graph of players and their connections. The players are represented by the nodes in the graph, with arcs representing connections between players. This idea ties closely to the networks that IAs may work across. Players with arcs between them interact with each other in rounds of The Prisoner's Dilemma. Nowak and May's [38] ealier work - which inspired Nowak's later paper [37] - did not give individual's any memory of past interaction.

This lack of memory limited Nowak and May to pure cooperators and pure defectors. In Nowak's book 'Evolutionary Dynamics' [36], his exploration of evolutionary graph theory and spatial games (chapters 8 and 9) showed that the shapes of the lattice linking the players and different concentrations of cooperators and defectors on those shapes has a great effect on the evolution of cooperation. Visualised in figures 5.3 and 5.4.

Nowak's [37, 36] and Nowak and May's [38] work on these games on graphs is limited in terms of strategies and also in terms of the fixed shape of its graphs. However, the work proves a key point: the structure of who interacts with whom can play a key role in supporting cooperation in large populations.

In real life, individuals will often mostly interact in their close social circles. For example, a Meerkat may interact with others of their family group, a drongo bird which calls to warn of predators, the predators and others who are geographically close to them. The graph in this case represents the close geographic ties.

I imagine the use of Nowak [37, 36] and Nowak and May's [38] work to employ a network not as a representation of a physical network structure or geography, but as a representation of the choices made by IAs with whom they wish to interact with.

This network would be a constantly changing and adapting network of IAs. The IAs would not concern themselves with the strategy they employ towards whom they are forced to interact with. Instead, their strategy is to select those whom they wish to interact with, thus, effectively constructing a graph of network reciprocation. How these changing graph connections would affect cooperation is unbeknownst to me. Indeed, whether Nowak's rules would still apply would be interesting to find out.

Nowak [36] found that some shapes supported cooperators in groups. Cooperators could make use of these shapes by deliberately forming them to protect one another. While another set of shapes were found as 'amplifiers' for evolution, maybe defectors could make use of these sorts of shapes to invade groups of cooperators.

I can see IAs having strategies as to how to build these shapes. However, an issue may arise in which cooperative agents find it hard to reach out to other cooperators which are not part of their current shape.As such, there may be possible prevention of the spread of cooperation, thus limiting these groups. However, this concept is worth investigating, and the problem could possibly be overcome using some kind of bridging mechanism.Compare to [21] organisational relationships.

	A	B
A	a	b
B	c	d

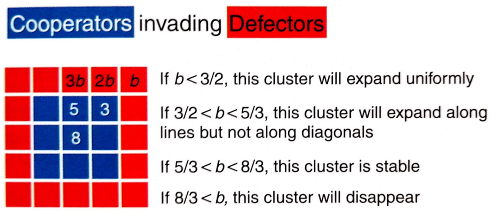


Figure 5.3: How can cooperators invade defectors? Taken from Nowak's book Evolutionary Dynamics [36]. The squares represent nodes and the players interact with the players to each side of them and diagonally. The value b is from the payoff matrix in table 5.1

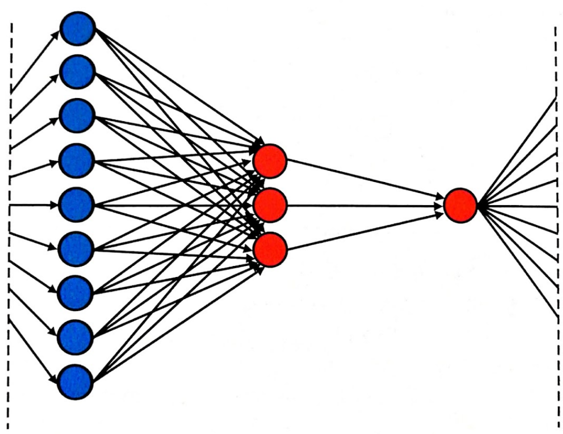


Figure 5.4: Shapes that can amplify selection include the funnel, the star and the super-star [36]

Table 5.1: The payoff matrix for when individuals interact. Cooperators are in blue and are called A, and defectors in red and called B. Taken from Nowak's book Evolutionary Dynamics [36].

5.1.3 Group Selection

Group selection is another mechanism described by Nowak [37]. This mechanism splits one population into multiple groups. Within these groups, The Prisoner's Dilemma is played and reproduction occurs which is proportional to each players payoff. If using the payoff matrix in table 2.2, then cooperators can work together to produce a payoff of three, while defectors can only produce five or two for both players in the interaction.

The group size increases until a certain point, at which the group may split. If the group does split (which is stochastically chosen) then another group is destroyed. The effect is multi-level selection.

Nowak found that due to the higher payoff between cooperators, reproduction will occur more quickly within the groups they dominate, than the groups filled with defectors. The faster the reproduction, the quicker the group size grows, making it more likely for groups of cooperators to survive while groups of defectors will shrink and be destroyed. These dynamics

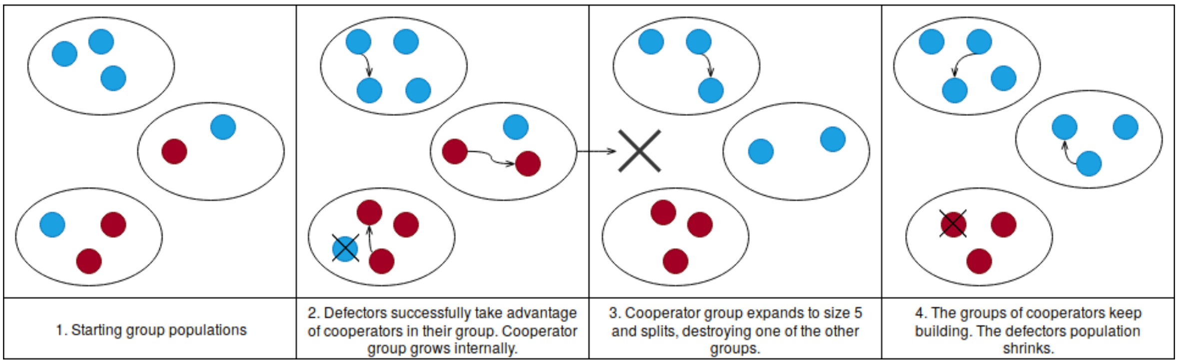


Figure 5.5: The dynamics of multi-level selection as described by Traulsen and Nowak [55].

are displayed in figure 5.5.

Traulsen and Nowak [55] limited themselves to cooperators and defectors, but noted that other strategies could be built into their model. A limitation to applying group selection to MASs is the groups themselves. In this model, individuals do not interact with individuals in other groups, creating a barrier between them.

Even our current networks span the globe and don't always have harsh barriers between them - even when security is high, these barriers can often be broken. Furthermore, IAs are often built to be of service to others and limiting them to being of service only to one group greatly reduces the service an agent can supply and limits the society as a whole . Finally, the group mechanics of splitting and destroying another group of individuals does not naturally match the paradigm of MASs.

I would suggest that this is not a mechanism which would be useful to apply to MASs unless you are modelling clusters of agents, with each cluster in competition. This idea is not the aim of my project; however it could be another interesting project to take up.

5.1.4 Alternate Agent Architectures

5.2 Theoretical Framework

5.2.1 Environment Properties

I have already discussed how the synchronised cycle steps makes my environment static, but what about Russell and Norvig's [45] 4 other properties. Knowing the intention of other agents is key to deciding on whether other agents should cooperate or not. However, agents cannot view all interactions, and gossip may be distorted so an agent may not know all details it needs to know for action decisions. As such, the environment I have delineated is only partially observable.

I also argue that the environment is deterministic from a generational point of view. Milinski *et al.* [29] claimed that individuals using the standing strategy aim for a good standing, and Leimar and Hammerstein [25] argue that the standing strategy is good as it allows individuals to punish bad individuals. However take 3 agents a, b, c and d. Agent b has defected previously against d who has a good standing according to a. Agent a then chooses to punish b but is still believing this won't reduce their standing. However, agent c did not observe b's defection against d but is an onlooker for c's defection against b.

From the proceedings the environment appears to be non-deterministic, as there is more than one outcome to an action. However, this is due to the partial observability of the environment not the determinism, as highlighted is possible by Russell and Norvig [45]. The current state (c's lack of knowledge on b) and the actions selected by a and b completely determine the next state of the system.

However, from a community point of view the game can be seen as non-deterministic due to the reproduction mechanism. This includes a chance of mutation and the actual selection of agents is stochastic. As such the actions of the agents cannot guarantee that their strategy will be highly propagated, the higher the fitness they get (which is decided by state and actions) affects their chances greatly but not fully.

The environment is also nonepisodic. To take a proof by contradiction approach suppose the environment is episodic. According to Russell and Norvig [45] an environment is episodic if subsequent episodes do not depend on what action occurs in previous episodes. Episodes consist of perceiving and then deciding. But from the percepts and actions subsections (3.2.3 and 3.2.4) we know that actions from one timepoint can generate percepts in the next. This is a contradiction and as such the environment is nonepisodic.

In summary the environment I have outlined in this section is inaccessible, deterministic from a generational point of view yet non-deterministic from a community point of view, nonepisodic, static and discrete.

5.2.2 Strategies

include list of strategy components and trust models, all associated together.