AN AGENT-BASED MODEL FOR EVOLUTION OF COOPERATION WITH PROACTIVE INFORMATION GATHERING

by

Nahid Mohammad Taheri

B.Sc., Applied Mathematics, University of Isfahan, Iran, 2011

THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE

UNIVERSITY OF NORTHERN BRITISH COLUMBIA

April 2018

© Nahid Mohammad Taheri, 2018

Abstract

This thesis proposes a new model to investigate the impact of proactive information gathering upon the evolution of cooperation among self-interested agents in a multiagent system. It builds upon an existing game-theoretical model of spatially distributed mobile agent population with energy-based individual life cycle, in which individuals keep playing one-shot Prisoner's Dilemma games in neighbourhood encounters. Using proactively gathered information about past behaviour of others, advanced agents in the new model can dynamically adjust their strategies towards different types of opponents. Simulation experiments establish some patterns of how this ability impacts the evolution of cooperation in the presence of varying levels of environmental adversity. The adequacy of the model is demonstrated through a specific design involving two types of advanced agents, one oriented towards cooperation, the other towards defection. The results show that cooperation prevails in a substantially larger area of parameter space than in the basic model without information gathering.

Contents

A	bstra	\mathbf{ct}	i
Li	st of	Figures	v
\mathbf{Li}	st of	Tables	ix
A	cknov	wledgements	x
D	edica	tion	xi
1	Intr	oduction	1
2	Bac	kground and Related Work	8
	2.1	Multiagent Systems (MAS)	8
	2.2	Game Theory	10
	2.3	Evolutionary Game Theory	15
	2.4	Evolution of Cooperation in MAS	16
	2.5	Evolution of Cooperation in Spatial Structure	17
	2.6	Evolution of Cooperation with Information Exchange	20
	2.7	Agent-Based Modelling	24

3	Pro	blem Statement		25
	3.1	The Motivation and Research	Objectives	25
	3.2	The IG-Model		29
		3.2.1 Advanced Agent		29
		3.2.2 Strategy Adjustment		30
	3.3	Cluster studies and neighbour	hood profile	32
4	Sim	ulator Architecture		33
	4.1	The Simulator Requirements		33
		4.1.1 Functional Requirement	nts	34
		4.1.1.1 Set and Run	Experiment	34
		4.1.1.2 Display Expe	eriment	36
		4.1.2 Non-functional Require	ements	37
		4.1.3 Domain-specific Requi	rements	37
	4.2	The Simulator Structure		39
		4.2.1 Controller		40
		4.2.2 MAS-layout		42
		4.2.3 Front End		44
	4.3	The System Behaviour		45
	4.4	Experimentation		57
5	Eva	luation		59
	5.1	Parameter Space		59
	5.2	The Basic Model		61
		5.2.1 Clusters and Neighbou	urhood Profiles	64
	5.3	The IG-Models		67
		5.3.1 Initial Agent Population	on	67

6	Con	clusio	ns and Future Work	92
		5.4.1	Results Validation	89
	5.4	Perfor	mance Comparisons	85
		5.3.4	IG-ACD	81
			5.3.3.1 Observations and Analysis	72
		5.3.3	IG-AD	71
			5.3.2.2 Clusters and Neighbourhood Profiles	69
			5.3.2.1 Observations and Analysis	68
		5.3.2	IG-AC	68

	Bib	liogr	aphy
--	-----	-------	------

95

List of Figures

2.1	Agent's life cycle as defined in the [Smaldino et al., 2013] 18
4.1	The use case diagram of the simulator
4.2	The high level structure of the simulator
4.3	The controller
4.4	The advanced agent architecture
4.5	The front end
4.6	The control panel
4.7	The mechanics of the <i>experiment</i>
4.8	The mechanics of the run
4.9	The mechanics of the <i>round</i>
4.10	The agent's activity diagram
4.11	The agent's state-machine representation
4.12	The GUI window
4.13	The GUI window, when we only have two types of agents in the pop-
	ulation
4.14	The Mouseover tool
4.15	The report viewer
4.16	The Cluster and its information

5.1	The number of cooperators for different values of S in the basic model.	61
5.2	The number of defectors for different values of ${\cal S}$ in the basic model	62
5.3	The number of defectors and cooperators in the long-run when S is -1	
	in the basic model	63
5.4	The number of cooperators for different values of cost of living in the	
	basic model	64
5.5	The number of defectors for different values of cost of living in the basic	
	model	64
5.6	An example of a dying cluster in the basic model	65
5.7	An example of a rising cluster in the basic model	65
5.8	An example of the number of agents and cooperators' neighbourhood	
	profile in a dying cluster in the basic model	66
5.9	An example of the number of agents and cooperators' neighbourhood	
	profile in a rising cluster in the basic model	66
5.10	The number of cooperators for different values of S in the IG-AC model.	69
5.11	The number of cooperators for different values of cost of living in the	
	IG-AC model	69
5.12	The number of neighbours in different types of agent's neighbourhood	
	in the IG-AC model.	70
5.13	Clusters' development in the IG-AC model	70
5.14	The number of cooperators for different values of S in the IG-AD0.3	
	model	72
5.15	The number of cooperators for different values of cost of living in the	
	IG-AD0.3 model	72
5.16	The number of cooperators for different values of S in the IG-AD0.5	
	model	73

5.17	The number of cooperators for different values of cost of living in the	
	IG-AD0.5 model	73
5.18	The number of cooperators for different values of S in the IG-AD0.7	
	$model \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots $	73
5.19	The number of cooperators for different values of cost of living in the	
	IG-AD0.7 model	74
5.20	The number of cooperators in the basic and the IG-AD models	74
5.21	The number of defectors in the basic and the IG-AD models $\ . \ . \ .$.	75
5.22	The number of advanced defectors and naive defectors in the IG-AD	
	models with a low punishing value of S	76
5.23	The number of advanced defectors and naive defectors in the IG-AD	
	models with a high punishing value of S \hdots	76
5.24	The number of cooperators in defectors' neighbourhood in the IG-	
	AD0.3 model	78
5.25	The number of cooperators in defectors' neighbourhood in the IG-	
	AD0.5 model	78
5.26	The number of cooperators in defectors' neighbourhood n the IG-AD0.7	
	model	79
5.27	The number of agents in a rising cluster in IG-AD0.3 model $\ . \ . \ .$.	79
5.28	The number of agents in a dying cluster in IG-AD0.3 model $\ . \ . \ .$.	80
5.29	An example of a rising cluster in the IG-AD0.3 model	80
5.30	An example of a dying cluster in the IG-AD0.3 model	80
5.31	An example of cooperators' neighbourhood profile in a dying cluster in	
	the IG-AD0.3 model	81
5.32	An example of cooperators' neighbourhood profile in a rising cluster in	
	the IG-AD0.3 model	81

5.33	The number of cooperators for different values of S in the IG-ACD0.3	
	model	82
5.34	The number of defectors for different values of S in the IG-ACD0.3 model	82
5.35	The number of cooperators for different values of cost of living in the	
	IG-ACD0.3 model	83
5.36	The number of cooperators for different values of S in the IG-ACD0.7	
	model	83
5.37	The number of cooperators for different values of cost of living in the	
	IG-ACD0.7 model	84
5.38	The number of advanced defectors and naive defectors in the IG-ACD	
	models	84
5.39	The number of advanced defectors and naive defectors in the IG-ACD	
	models	84
5.40	The outcomes of the evolution in the basic model $\ldots \ldots \ldots \ldots$	87
5.41	The outcomes of the evolution in the IG-ACD0.3 model	87
5.42	The outcomes of the evolution in the IG-AD0.3 model	89
5.43	The outcomes of the evolution in the IG-AC model	89
5.44	An example of compared results with error bars	91

List of Tables

2.1	The payoff matrix for the PD. The PD game is defined when this con-	
	dition holds: $T > R > P \ge S$. In addition, when the agents play	
	repeatedly, $2R > T + S$ is a requirement	13
2.2	The payoff matrix for a symmetric game	13
2.3	The payoff matrix for the Prisoner's Dilemma game, where $g,l>0,g<$	
	l+1 [Heller and Mohlin, 2016]	23
2.4	The payoff matrix for the Defensive Prisoner's Dilemma game, where	
	1 = g < l = 3 [Heller and Mohlin, 2016]	23
2.5	The payoff matrix for the Offensive Prisoner's Dilemma game, where	
	2.3 = g > l = 1.7 [Heller and Mohlin, 2016]	23
2.6	The payoff matrix for the Mild Prisoner's Dilemma game, where $g =$	
	l = 0.2 < (l+1)/2 = 0.6 [Heller and Mohlin, 2016]	23

Acknowledgements

I would like to sincerely express my gratefulness to my supervisor, Dr. Jernej Polajnar, for his support, advice and patience throughout all stages of this thesis.

This Thesis is dedicated to my husband

Mani Samani

for his endless love and support

Chapter 1

Introduction

Cooperation is often considered as one of the key concepts of multiagent systems (MAS) [Doran et al., 1997] [Buccafurri et al., 2004] [Salazar et al., 2011]. An agent is an entity situated in an environment, and it is able to perform intelligent and autonomous action in order to achieve its delegated goals [Wooldridge, 2009]. A multiagent system consists of multiple agents interacting with each other, which are assigned individual and team-level tasks. Self-interested agents in a multiagent system, whether designed by a single team or independently, may need to cooperate with each other to perform their tasks and advance the overall objectives of the system [Kraus, 1997] [Jennings, 1995].

The designers of agents typically have two different ways to design a multiagent system. The first option is to design agents with fixed cooperation patterns. The alternative way is designing each agent with an ability to develop cooperative behaviour through interactions with other agents. The latter way may prove more productive, especially in a dynamic environment where agents need to adjust their behaviour to the changes that are difficult to predict. In addition, the second approach applies more naturally to societies composed of independently designed agents. To design agents with the ability to develop cooperation proactively, it is helpful to understand how cooperation can evolve in agent societies under different circumstances.

The groundwork for studies of the evolution of cooperation among self-interested individuals has been provided by Hamilton [1963], Axelrod [1984], Smith [1982], and Nowak [2006]. Numerous researches in the evolution of cooperation among selfinterested agents are based on the *evolutionary game theory* [Weibull, 1997] [Gintis, 2000] [Colman, 2013].

Studies in the evolution of cooperation, that are rooted in evolutionary game theory, are typically built on the *Prisoner's Dilemma (PD)* game [Shoham and Leyton-Brown, 2009]. The PD game is representative of the wider approach to the study of cooperation between self-interested agents, contrasting two extreme types of behaviour: characterized as 'selfish' (defection) and 'altruistic' (cooperation). A cooperator helps other individuals at its own cost. A defector does not pay any cost and takes advantage of cooperators.

Among agents playing the PD game, individuals engage in pairwise interactions, and each of them can select to cooperate or to defect. If both players cooperate (mutual cooperation), they will receive a higher payoff than when they both defect (mutual defection). However, if one cooperates and the other one defects, the defector gets the highest payoff, and the cooperator gets the lowest. In the one-shot PD game, regardless of what the other player does, defection is always the best strategy because it has the highest payoff. As a result, even though mutual cooperation is more profitable than mutual defection, self-interested agents always defect in one-shot PD game to receive the highest payoff [Elkind and Markakis, 2013]. However, several studies prove that the consistent defection is not always the best strategy. Different mechanisms have been developed to promote the evolution of cooperation in the agent societies [Doebeli and Hauert, 2005].

In some of the models studied, an agent that needs to decide its strategy in the next encounter could benefit from having some information about the profile of other agents. It seems that in most of these models, cooperation can evolve if self-interested individuals can obtain some information about other agents, such as the history of their interactions and their visited opponents [Bachi et al., 2010] [Takahashi, 2010] [Awaya, 2014] [Heller and Mohlin, 2016]. An agent then can adjust its strategy based on the obtained information.

This thesis focusses on the impact of dynamically adjustable agents' strategies, based on the proactive gathering of information about the behaivor of other agents, upon the evolution of cooperation within an existing model of agent societies. The model that we adopt as the basis of our research is formulated by Smaldino et al. [2013]; henceforth, we refer to it as the *basic model*.

The basic model explores the impact of environmental adversity, or harsh environment, characterized by the *cost of living* and the *unreciprocated cooperation cost*, upon the evolution of cooperation in a population of agents. In that model, the agents are situated on a toroidal grid. The agents have energy-based life cycle and the population evolves through the births and deaths. Each individual agent can interact with a randomly chosen partner from its spatial neighbourhood by playing the PD game. If no neighbour is available as a partner, the agent can move to an empty adjacent cell. Each agent is given a starting energy value, which is subsequently modified by the game payoffs. In addition, the energy of each agent is reduced in each round by a fixed amount representing the cost of living. Smaldino et al. [2013] show that in moderately harsh environmental conditions, cooperation tends to prevail over selfish behaviour eventually. They also present the circumstances under which cooperation evolves in the basic model. This observation has raised the research question of how that development is affected by allowing agents to dynamically adjust their strategies based on proactively collected knowledge of past behaviour of other agents.

In this thesis, we approach that question by adopting and extending the premise of the basic model. In the basic model, the population consists of two types of agents: cooperators, which always cooperate, and defectors, which always defect. We retain those agent types in our model and call them *naive* cooperators and *naive* defectors respectively. In addition, we introduce *advanced* cooperators, which favour cooperative behaviour, and *advanced* defectors, which favour selfish behaviour, but are capable of pursuing more complex, dynamically adjustable strategies. Advanced agents are capable of collecting some information about their opponents, when they play with them. The collected information is stored in their personal game memory. Also, they can inquire information of past behaviour of other agents in the population.

Advanced agents' strategies are adjusted dynamically based on their personal game memory and the proactively gathered knowledge of past behaviour of other agents. In the presence of some knowledge about the past behaviour of the opponent, an advanced agent may either cooperate or defect, as dictated by its advanced strategy, motivated by its tactical approach to perceived cooperators or defectors. However, in the absence of any such information, advanced cooperators always cooperate, and advanced defectors always defect.

In this context, we investigate the information gathering (IG) techniques and advanced strategies that facilitate the evolution of cooperation in the society in spite of the presence of defectors. In particular, we explore whether proactive information gathering can lower the harshness threshold of environmental adversity at which cooperation evolves compared to the basic model. One should note that the hypothesis that proactive information gathering may facilitate the success of advanced cooperators requires careful study, as advanced defectors can also use the gathered information to promote their own success ¹.

Within this context, we design IG-models with specific strategies of advanced agents, provide a rationale for that particular design, perform a series of experiments across a space of parameters representing varying levels of environmental adversity, and compare the outcomes with respect to evolution of cooperation with the outcomes of the basic model over the same parameter space. The results confirm that, compared to the basic model, the presence of advanced agents with IG capabilities significantly extends the region of parameter space in which cooperators prevail over defectors.

After the information is obtained by an advanced agent, it can be used in different ways. For example, advanced cooperators can act conditionally based on their opponents' previous actions as represented in [Nowak and Sigmund, 1998] where each individual only cooperates with those that cooperated in their previous interactions. The agent's strategic reasoning is an open research dimension, which can be more explored during the further experimentation for future works.

For modeling and simulation, we develop a simulator that supports the concurrent simulation of multiple agents interaction in an environment. The simulator provides an interactive framework for comparing the number of different types of agents and population development, across the distinct parameter spaces. It includes facilities for

¹The impact of strategy adjustment based on information gathering by advanced agents to the development of cooperation is being studied by our research group in the context of different models of agent society. An ongoing study parallel to this thesis uses as basis a substantially different game-theoretic model of evolving agent society [Nyantakyi, 2018].

the study of localized evolutionary phenomena, such as neighbourhood profiles and cluster development. The IG-models are investigated through simulation experiments, and the impact of proactive information gathering upon the evolution of cooperation is determined in comparison with the basic model.

The main contributions of this thesis are as follows. First, a concrete model of agent society has been constructed in which the designs and strategies of informationgathering advanced agents can be examined with respect to their impact upon the development of cooperation under moderately harsh environmental conditions, extending the modeling studies in evolution of cooperation in [Smaldino et al., 2013]. Second, concrete designs of advanced cooperators and advanced defectors have been introduced, implemented, and studied through simulation experiments. The results show that the presence of advanced agents significantly extends, in comparison to the basic model, the region of the parameter space representing the harshness of living conditions (as represented by the cost of living and the payoff for unreciprocated cooperation) in which cooperators eventually prevail over defectors. Third, our experimental studies of the specific model with advanced cooperators and defectors, as well as auxiliary models with advanced agents of a single type, in our view demonstrate the adequacy of the general model as a vehicle for further study of specific IG-models and their impact upon the development of cooperation, possibly including comparisons with results and insights from parallel developments of IG-models built upon basic models of different structure.

The rest of this thesis is organized as follows. Chapter 2 covers the background and related work. Chapter 3 describes the motivations and research objectives and formulates the main objectives in the IG-models. Chapter 4 lays out the simulator architecture, including the simulator requirements, structure, and behaviour. Chapter 5 elucidates evaluation methods to analyze the results of simulation experiments, and Chapter 6 discusses the conclusions and future work.

Chapter 2

Background and Related Work

In this chapter, we review several groundwork studies and combine those with related recent research in multiagent systems, evolutionary game theory, evolution of cooperation in multiagent systems, evolution of cooperation in spatial structure, evolution of cooperation with information exchange, and agent-based modelling, with the focus relevant to this thesis.

2.1 Multiagent Systems (MAS)

Wooldridge [2009] defines an intelligent agent as "a computer system that is situated in some environment, and that is capable of autonomous action in this environment to meet its delegated objectives." Shoham and Leyton-Brown [2009] describe a multiagent system (MAS) as "a system that includes multiple autonomous entities with either diverging information or diverging interests or both." Wooldridge and Jennings [1995] suggest some capabilities for an agent to be considered as intelligent. An intelligent agent is expected to be *reactive*, i.e., able to perceive the environment and respond to its changes, *proactive*, i.e., able to have goal-directed behaviours, and *social*, i.e., able to interact with other agents and possibly humans.

The environment in which an agent is situated is generally classified according to four criteria [Russell and Norvig, 1995] explained in the following:

- Accessible versus inaccessible: In an accessible environment, the agent can obtain complete information about the environment; however a real-world environment, such as the physical world, is mostly inaccessible.
- **Deterministic versus non-deterministic:** In a deterministic environment, each agent's action has a single guaranteed effect on the resulting environment state.
- **Static versus dynamic:** In a static environment, the state of the environment can only change by an agent's action; however, in a dynamic environment, other processes influence the environment beyond the agent's control.
- **Discrete versus continuous:** In a discrete environment, there is a fixed and finite number of actions and percepts. Otherwise, an environment is called continuous.

In the modelling of agents interacting with their environment, the environment is initially in some state. While an agent is interacting with the environment, it perceives the current state of the environment and performs an action. The environment can transit to another state in response to the performed actions. The states of the environment can be correlated with some real values, know as *utility*. This real value indicates how good each state is and allows agents to decide what to do. Practical reasoning [Bratman, 1987] is a particular model of decision making directed towards actions, whereas the *theoretical reasoning* is directed towards beliefs. In human practical reasoning, there are two different processes for decision making called *deliberation* and *means-ends reasoning*. Deliberation is to answer the question of what desired state we want to achieve, and the means-ends reasoning is to decide how to achieve the desired state. The practical reasoning agents are designed with the mental states similar to the ones in the human mind. These mental states are used in the agents' decision making processes. The output of deliberation is *intention*, and the output of means-ends reasoning is *plan*.

The best known practical reasoning framework is the *Belief-Desire-Intention (BDI)* proposed by Bratman [1987] and formulated by Shoham [1993]. In the context of BDI framework, the agents are considered rational with mental states of belief, desire, and intention. Beliefs are information the agent has about the environment; desires are all the possible states of affairs that the agent might like to accomplish; and intentions are the desired states that the agent has decided to work towards. Russell and Norvig [2009] define a rational agent as "one that acts so as to achieve the best outcome or when there is uncertainty, the best expected outcome." In BDI, a rational agent is committed to its intention, as long as the intention is valid. The conceptual framework of the *BDI* model is described in [Bratman et al., 1988].

2.2 Game Theory

Generally, there are two kinds of MAS: *cooperative* and *non-cooperative*. In the cooperative systems, the agents share a common goal, however, in the non-cooperative systems, the agents are self-interested. Although the self-interested entities have their own desires, they still interact with other agents in the system. The best framework for studying and analyzing the non-cooperative MAS is the main branch of *game theory*, called *non-cooperative game theory*. The non-cooperative game theory is mainly used to model the behaviour of self-interested entities and their associated decision making in different situations [Elkind and Markakis, 2013].

The other branch of game theory is called *cooperative game theory*, where the main modelling entity is the group of self-interested individuals [Shoham and Leyton-Brown, 2009]. Since in this study, we are mostly interested in observing the behaviour of each individual, we only cover the non-cooperative game theory in this background.

The theory of games was first formalized by von Neumann and Morgenstern [1944] to analyze human economic behaviour. In the game theory, a game is defined as an interaction among multiple agents with different preferences over the states of the environment, and each agent behaves rationally.

Utility theory is the main way to capture an agent's preferences. As Shoham and Leyton-Brown [2009] define, "[utility theory] aims to quantify an agent's degree of preference across a set of available alternatives. The theory also aims to understand how these preferences change when an agent faces uncertainty about which alternative he will receive." Considering the utility-theoretic assumptions the *utility function* represents an agent's preferences by assigning a numeric value to each state of the environment. A rational agent always wants to maximize its own utility. In an environment that consists of multiple agents, the game theory is used to describe the strategic interactions among self-interested agents, where each individual agent wants the maximum utility.

One of the most common representations of the strategic interactions between

agents in the game theory is the *normal-form* game [Shoham and Leyton-Brown, 2009]. A normal-form game represents each agent's (player) utility for different states of the environment, where the states of the environment only depend on the players' combined actions [Shoham and Leyton-Brown, 2009].

Definition 2.1.1 (Normal-form game): A (finite, *n*-person) normal form game is a tuple (N, A, u) where:

- N is a finite set of n players, indexed by i.
- $A = A_1 X... X A_n$, where A_i is a finite set of actions available to player *i*.
- $u = (u_1, ..., u_n)$, where u_i is a real-valued utility function for player *i*.

The utility function in the Definition 2.1.1 is also called the *payoff* function. The payoff function assigns a real number to a combined-action profile for each agent. The payoff values are shown in a matrix, called the *payoff matrix*. In a payoff matrix, each row represents a possible action for player 1, each column represents a possible action for player 2, and each cell represents a possible outcome. The payoff values for the corresponding outcome are shown as a pair of (X_1, X_2) , where X_1 is the payoff to player 1, and X_2 is the payoff to player 2. The *Prisoner's Dilemma (PD)* game is an example of a normal-form game.

Definition 2.1.2 (Prisoner's Dilemma): The name of this game derives from the interpretation of two suspects, who are accused of committing a crime together. They reside in different cells and have no way of communicating. Each has the option of either confessing to the offense, *defection* (with respect to the other player) or denying the charge, *cooperation* (with respect to the other player). If both cooperate (mutual cooperation), they will go to jail for one year. If both defect (mutual defection), each of them will go to prison for three years. If one cooperates and the other defects, the defector will be freed, and the cooperator will be jailed for four years [Elkind and Markakis, 2013].

The values in the PD payoff matrix are defined as R, for the mutual cooperation, and P, for the mutual defection. When one player cooperates and the other defects, the defector gets the highest payoff, T, and the cooperator gets the lowest, S.

		Playe	er 2
		Cooperation	Defection
Plavor 1	Cooperation	(R,R)	(S,T)
I layer I	Defection	(T,S)	(P, P)

Table 2.1: The payoff matrix for the PD. The PD game is defined when this condition holds: $T > R > P \ge S$. In addition, when the agents play repeatedly, 2R > T + S is a requirement.

A normal-form game can be either *symmetric* or *asymmetric*. The payoff for the players does not depend on their identities in a symmetric game; it depends on the interaction. Also, the players do not have different roles in the game. PD game is an example of a symmetric game.

Definition 2.1.3 (Symmetric 2X2 Game): A two-player normal-form game is called a symmetric game if it has the following form:

Strategy	А	В
А	(x, x)	(u, v)
В	(v, u)	(y, y)

Table 2.2: The payoff matrix for a symmetric game

According to Definition 2.1.1, each agent has a set of available actions. The way that an agent decides what action to perform is called *strategy*. One way to play a game is to choose a single action. This strategy is called the *pure strategy*. In a single PD game, the defection is always the best strategy because it has a higher payoff, regardless of the other player's action. An agent can also use some probability distribution to choose randomly over its set of available actions. This kind of strategy is called the *mixed strategy* [Shoham and Leyton-Brown, 2009]:

Definition 2.1.4 (Mixed Strategy): Let (N, A, u) be a normal-form game, and for any set X let $\Pi(X)$ be the set of all probability distribution over X. Then the set of mixed strategies for player i is $S_i = \Pi(A_i)$.

In the single-agent situation, we use the *optimal strategy* notion, which refers to the strategy that maximizes an agent's utility. In the multiagent setting, the concept of optimal strategy is not useful anymore since the best strategy for a given agent depends on the actions of the other agents in the environment. In that case, game theorists define different *solution concepts* [Shoham and Leyton-Brown, 2009]. One of the most important notions of solution concepts in the game theory is the *equilibrium* in general, and the *Nash equilibrium* in particular.

From one individual agent's perspective, if an agent knows the other agent's strategy, it just needs to select an action that maximizes its utility. That means an agent needs to determine its *best response* to the strategy profile $s_{-i} = (s_1, ..., s_{i-1}, s_{i+1}, ..., s_n)$ [Shoham and Leyton-Brown, 2009]. The strategy profile s_{-i} includes all the agents' strategies, but not *i*.

Definition 2.1.5 (Best Response): Player *i*'s best response to the strategy profile s_{-i} is the mixed strategy $s_i^* \in S_i$ such that $u_i(s_i^*, s_{-i}) \ge u_i(s_i, s_{-i})$ for all strategies $s_i \in S_i$.

In general, an agent will not have any information on what strategies the other

agents would play. However, the idea of best response can be used to define the Nash equilibrium [Shoham and Leyton-Brown, 2009].

Definition 2.1.6 (Nash equilibrium): A strategy profile $s = (s_1, ..., s_n)$ is a Nash equilibrium if s_i is a best response to s_{-i} , for all agents i, and $i \in N$, the finite set of n players.

A Nash equilibrium is a *stable* strategy profile. In a stable strategy profile, none of the agents would prefer to change their strategy if they know the selected strategy of the other players.

2.3 Evolutionary Game Theory

Evolutionary game theory (EGT) was developed by Smith and Price [1973] as an extension of game theory for modelling the evolution of life forms. The focus of EGT is on the population of agents playing a game against random opponents, all of whom are competing. The competition among individuals results in selecting agents with the higher utility. Therefore, the agents want to adopt the most successful strategy in terms of payoff. EGT is used to aid analysis of evolutionary selection in an interactive environment [Weibull, 1997] [Tuyls and Tumer, 2013].

The main assumption of the EGT is that in a population of an infinite number of players playing a game, all the agents play the same strategy initially. However, there is a chance that a small number of agents begin to play the game using a new strategy. The evolutionary success of a strategy is measured by the core equilibrium concept of the EGT, called the *evolutionary stable strategy (ESS)*. If the payoff obtained by the

new strategy is smaller than the payoff obtained by the old strategy, the new strategy will disappear after some generations, and the original strategy becomes evolutionary stable.

Definition 2.3.2 (Evolutionary Stable Strategy): Given a symmetric twoplayer normal-form game (G=1,2, A, u), and a mixed strategy S, we say that S is an ESS if and only if for some $\epsilon > 0$ and for all other strategies S':

$$u(S, (1-\epsilon)S + \epsilon S') > u(S', (1-\epsilon)S + \epsilon S')$$

where u is referring to the utility function.

EGT is used in different fields such as biology [Sigmund and Nowak, 1999] and sociobiology [Wilson and Wilson, 2007], and also to study the evolution of cooperation among self-interested agents [Axelrod and Hamilton, 1981].

2.4 Evolution of Cooperation in MAS

Designing and building systems of intelligent and self-interested agents that can work together is a complex task in computer science [Kraus, 1997]. Agents should perform individual and team-level tasks and need to cooperate with each other to satisfy a joint goal and increase the system's efficiency. Designers can build agents with the fixed cooperative patterns. However, the agents may need to develop cooperation at the run time, especially when they are designed independently and are located in an unknown dynamic environment. Therefore, the agent designers may benefit from understanding how cooperation can evolve among self-interested agents. To address that subject, many game-theoretical models have been developed [Pendharkar, 2012]. Several studies have formulated the evolution of cooperation among the agents in graphs and social networks, [Szabó and Fath, 2007] [Floría et al., 2009]. Many researchers have tried to find certain circumstances which may or may not favour the cooperation, such as harsh environment [Smaldino et al., 2013], spatial structure [Majeski et al., 1997], variant group size [Barcelo and Capraro, 2015], and so on. Furthermore, different strategies have been developed to allow cooperators to interact more with each other, such as partner selection mechanisms and conditional movement [Aktipis, 2011], walk away strategy [Aktipis, 2004], and different movement patterns [Smaldino and Schank, 2012].

2.5 Evolution of Cooperation in Spatial Structure

Spatial structure is often considered as an effective factor of social evolution [Epstein et al., 1998]. In the evolutionary games with spatial structure, agents' interactions are limited to defined neighbours in the game space. That limitation can favour the evolution of cooperative behaviour [Brauchli et al., 1999] [Nowak et al., 2010].

We adopt the *basic* model for this thesis from [Smaldino et al., 2013], which studies the evolution of cooperation among self-interested mobile agents in a dynamic socio-spatial structure. The basic model investigates the impact of two factors on the evolution of cooperation, namely the *cost of unreciprocated cooperation* and the *environmental cost of living*. Unreciprocated cooperation cost refers to the payoff S of cooperation in the PD game, where one player cooperates and the other one defects.

Figure 2.1 shows an agent's life cycle in the basic model. In the basic model, the

agents are given an energy value. The agents' energy level is affected by the cost of living and game payoffs. They move to find a co-players and gain or lose resources, i.e., energy, through game interactions. The basic model is explained in the following paragraphs.



Figure 2.1: Agent's life cycle as defined in the [Smaldino et al., 2013].

• The Population

The population consists of agents a_i , $i \in I$, where I is a set of positive integers. The initial population is determined by the initial value of set I, and it is divided to the equal number of cooperators and defectors. The population evolves in discrete steps, where the set of agents can change through births and deaths. The population's capacity is capped at a constant number, the maximum carrying capacity, so the population cannot grow beyond its maximum carrying capacity.

• The Environment

The mobile agents are located randomly on a $q \times q$ discrete square grid with periodic (toroidal) boundaries, which resembles a torus [Oxford University Press, 2017]. Only one agent can occupy a cell. Each agent has its own neighbourhood, which includes the eight closest cells. Two agents are called neighbours if they are located in the same neighbourhood.

• The Interactions and Movement

Agents interact with each other by playing a 2-player game. Each agent can only play with its neighbours. The agent must find a co-player that is not engaged in another interaction. Each agent is given a certain unchangeable strategy (action) to play the game. If there is no opponent, the agent attempts to move to another cell in its neighbourhood. The movement is successful if there is an empty cell in the agent's neighbourhood.

• Experiments and rounds

The agents' interactions and movement occur in the discrete rounds. The basic model is built with the MASON [Luke et al., 2005], a discrete event multiagent simulation toolkit, which allows for scheduling agents to run every time step or round.

• The Energy and Population Dynamics

Each agent in the initial population has a starting energy value. That value is assigned randomly by the uniform distribution (U[a, b]), where a is the minimum and b is the maximum). Agents should be able to accumulate energy to remain alive and to reproduce. The cost of living and game payoffs affect the energy of agents; however, an agent's energy level is always capped at a constant amount, the maximum energy. Furthermore, an agent would be removed from the game space if its energy is below a certain value called the *death threshold*. Also, an agent can attempt to reproduce into a neighbouring cell if its energy reaches a certain value called the *birth threshold*. The birth is not successful if the population is at its maximum capacity or if there is no empty cell in the agent's neighbourhood. A new born agent receives a fixed starting energy from its parent, which results in the parent's energy level deduction.

• The Game

The two-player game used in the basic model is the PD game.

• Results

Under certain assumptions of the basic model, Smaldino et al. [2013] conclude the following results:

- When the cost of living has a fixed value and the cost of unreciprocated cooperation varies, the higher cost of unreciprocated cooperation results in losses for cooperators at the beginning; however cooperative agents can survive in the long run. By increasing the cost of unreciprocated cooperation, it is more difficult for cooperators to keep growing cooperative clusters.
- 2. When the cost of unreciprocated cooperation is constant, and the environmental cost of living takes different values, the higher cost helps cooperators in the long run, unless the cost of living is too high for any individual to survive.

2.6 Evolution of Cooperation with Information Exchange

Among all different frameworks for studying the evolution of cooperation in MAS, there are some models which emphasize the impact of having information about other individuals on the evolution of cooperation through the use of *reciprocity*, *social in*- formation, and trust and reputation [Mohtashemi and Mui, 2003] [Sabater and Sierra, 2002] [Sekaran and Sen, 1995].

The first well-known work studying evolution of cooperation with the knowledge of the opponent's past behaviour is the Axelrods tournament, where agents engage in a long sequence of repeated PD interactions with the same opponent, and they can decide whether to cooperate or defect based on the previous action of their opponent. The winning strategy of Axelrods tournament was *tit-for-tat* (TFT), in which an agent would cooperate the first time, and then it would repeat the action of its opponent from the previous round. If there are other cooperators in the tournament with which to interact, an agent with TFT strategy can increase its total score [Axelrod, 1984].

Sekaran and Sen [1995] investigate the concept of reciprocity and its impact on developing cooperative behaviour in a MAS with the use of probabilistic decisionmaking algorithms. Each agent has a number of tasks to perform. An agent can perform its tasks individually, or it can request help. The agent a will cooperate with the agent b if it seeks the future benefits from the agent b. Thus, the agent a uses the information about past interactions with the agent b to decide if it desires help or not. The result of that study shows that the cooperative agents perform better than others.

Evolutionary biologists, Nowak and Sigmund [1998], develop a model of indirect reciprocity by image scoring. In that model, each player has an image score. A player's image score can be modified locally by the other player engaged in the interaction and a few random observers. The image score of a cooperator will increase because of its altruistic act. Therefore, an informed player only cooperates with those that are likely to help others. The result suggests that when information is localized, cooperation can evolve based on the players' reputation in the system. The fact that an agent can cooperate to receive help from others agents in the future, results in developing some models with *social information* and *trust and rep-utation*. Mohtashemi and Mui [2003] claim that rather than observation and direct interaction with other agents, information can be gathered by trusting agents in a social network. In their model, agents can communicate and acquire information selectively from their social network. The results suggest that cooperation can evolve if the collected benefit from reliable agents in the network out-weigh the cost of helping untrustworthy individuals.

Sabater and Sierra [2002] use a trust-net to consider an agent's social interaction with others. In that model, the agents calculate some reputation values by analyzing other agents' social network. As Sabater and Sierra [2002] argue, the social network analysis can be used in different experimentations in real e-commerce environments.

[Heller and Mohlin, 2016] propose a model in which the agents play one-shot PD game with a random opponent and can observe a random sample of the previous behaviours of their opponents and decide whether to defect or cooperate. There are two main assumptions in their study. First, there is a particular group of committed agents, and they use a certain type of strategy. Second, all the individuals in the population do not start to interact at the same time. That study presents three main results based on the different variations of the PD game with respects to the equilibrium. First, in the offensive PD game, Table 2.5, defection is the unique equilibrium. Second, in the defensive PD game, Table 2.4, cooperation can remain stable in the presence of a combination of strategies, where all the agents defect if at least two defections occur in the opponent's history and cooperate when there is no defection. There are also a few agents, that defect in the case of a single defection in in the opponent's history. Third, in the *mild PD* game, Table 2.6, cooperation can

be sustained, if an agent can also acquire information about the behaviour of past opponents against the current co-player.

		Playe	er 2
		Cooperation	Defection
Playor 1	Cooperation	(1, 1)	(-l,1+g)
I layer I	Defection	(1+g,-l)	(0,0)

Table 2.3: The payoff matrix for the Prisoner's Dilemma game, where g, l > 0, g < l+1 [Heller and Mohlin, 2016].

		Player 2	
		Cooperation	Defection
Player 1	Cooperation	(1, 1)	(-3,2)
	Defection	(2, -3)	(0, 0)

Table 2.4: The payoff matrix for the Defensive Prisoner's Dilemma game, where 1 = g < l = 3 [Heller and Mohlin, 2016].

		Player 2	
		Cooperation	Defection
Player 1	Cooperation	(1,1)	(-1.7, 3.3)
	Defection	(3.3, -1.7)	(0, 0)

Table 2.5: The payoff matrix for the Offensive Prisoner's Dilemma game, where 2.3 = g > l = 1.7 [Heller and Mohlin, 2016].

		Player 2	
		Cooperation	Defection
Player 1	Cooperation	(1, 1)	(-0.2, 1.2)
	Defection	(1.2, -0.2)	(0, 0)

Table 2.6: The payoff matrix for the Mild Prisoner's Dilemma game, where g = l = 0.2 < (l+1)/2 = 0.6 [Heller and Mohlin, 2016].

[Takahashi, 2010] theoretically show that cooperation can be sustained, by an equilibrium, in a large population of players playing the repeated PD game. Each player is aware of its own past actions. In addition, each player can observe its current opponent's past actions without any cost. Players then choose a mixed strategy based on the available information. [Awaya, 2014] construct an equilibrium, which remains robust when players can choose to observe their current opponent's past action with

a very low cost.

[Bachi et al., 2010] investigate the role of communication in the emergence of cooperation in one-shot PD game. In that model, players talk to each other before playing the game, and they may learn the other player's strategy with a probability. The probability of receiving the information is the same for both players. The results show the levels of cooperation development under different conditions.

2.7 Agent-Based Modelling

Agent-based modelling (ABM), is a very powerful tool that has been used to simulate real-world problems and human behaviours in recent years [Helbing, 2012] [Bonabeau, 2002]. Several studies have used this technique to explore the evolution of cooperation among self-interested agents [Smaldino and Schank, 2012] [Smaldino and Schank, 2012] [Wilson and O'Brien, 2009] [Aktipis, 2004] [Epstein et al., 1998].
Chapter 3

Problem Statement

This chapter delineates the motivation for designing a new model for the evolution of cooperation among self-interested agents with proactive information gathering. This model is designed to investigate the combined effect of two principles, namely repeated interaction with knowledge of opponents' past behaviour and environmental adversity controllable by the experimenter, upon the evolution of cooperation. Section 3.1 describes the research motivations, and Section 3.2 presents the new model with proactive information gathering.

3.1 The Motivation and Research Objectives

Building multiagent systems (MAS), in which self-interested agents are able to perform a common task jointly, can be challenging since agents may need to cooperate to achieve their individual goals and improve the system performance. In fact, cooperation is considered as one of the key concepts of agent societies [Doran et al., 1997] [Buccafurri et al., 2004] [Salazar et al., 2011]. There are two possible ways to enable agents developing cooperation patterns. If the multiagent system is created by a single team and the environment is known, then cooperative behaviour can be built into agents at the design time. However, if agents are designed independently and located in an unknown environment, it might be beneficial to understand how cooperation can evolve through interactions among self-interested agents.

To formalize the evolution of cooperation among self-interested individuals, many studies have developed different game-theoretical models based on repeated pairwise encounters in which agents engage in a noncooperative two-player game [Gintis, 2000] [Colman, 2013]. In each encounter, agents play a game which allows both cooperative and selfish strategies, typically the Prisoner's Dilemma (PD) game [Shoham and Leyton-Brown, 2009], with the agents' cumulative payoff from the series of encounters representing individuals' success.

Cooperation and defection are the only actions available to an agent playing the PD game. As described in Section 2.2, in the one-shot PD game, no matter what the other player does, defection is always the best strategy because it has a higher payoff. However, it has been proven that consistent defection is not always the best strategy among n agents (n > 2) playing the PD game repeatedly as we can see in Axelrod's tournament [Axelrod, 1984]. In addition, some research studies show that evolution of cooperation is affected by several factors other than individual strategy choices, such as different population structure and the environment [Axelrod, 1997] [Majeski et al., 1997] [Barcelo and Capraro, 2015].

Our specific investigation is situated in the context of a game-theoretic model of an agent society with an energy-based life cycle within a spatially structured population and a controllable level of environmental adversity introduced by Smaldino et al. [2013] as discussed in Section 2.5. Smaldino et al. [2013] show how the impact of a harsh environment upon the evolution of cooperation among self-interested agents can be demonstrated in the context of game-theoretic agent-based population modelling. The environmental adversity is represented by the environmental cost of living and the cost of unreciprocated cooperation (corresponds to the value of S in the payoff matrix). A moderately high level of adversity favours cooperation in the long run, and the cooperative behaviour will develop. When the environment is too harsh, neither cooperators nor defectors can survive. Throughout this thesis we refer to that model as the *basic* model.

Agents in the basic model play with randomly selected opponents from their spatial neighbourhood, and they always play the strategy of pure cooperation or pure defection without having any information about other agents in the population. However, observations of animal species and human societies indicate that individuals use information, produced by the behaviour of other individuals, to facilitate their decision making, which can affect the cultural and biological evolution [Danchin et al., 2004] [Frith and Frith, 2012]. In addition, a variety of research studies have proved that cooperation can evolve if self-interested agents can have some partial knowledge about the past behaviour of their opponents [Bachi et al., 2010] [Awaya, 2014] [Heller and Mohlin, 2016].

Allowing agents to have some information about the behaviour of their opponent, while they are engaged in repeated interactions, was first studied in the Axelrod's tournament [Axelrod, 1984]. In that tournament, an agent plays a series of PD games against the same opponent, and it only has information about that particular opponent. Observation of the Axelrod's tournament and the basic model, has given rise to the research question of whether the combination of harsh environment and repeated interaction with partial knowledge about the past behaviour of opponents can facilitate the evolution of cooperation in agent societies.

In this study, we explore how the evolution of cooperation is affected by agents that are enabled to base their strategies on proactively gathered information about past cooperative or selfish behaviours of other agents by enhancing the basic model. In particular, we investigate if proactive information gathering can change the level of environmental harshness under which cooperation develops compared to the basic model.

In this study, we investigate the impact of having partial knowledge about the behaviour of other agents on the evolution of cooperation in MAS. This new model intends to empower the cooperators' success by enriching the basic model with a new element of information acquisition. Developing cooperative behaviours among self-interested agents in this research can be applied to more complex systems, where agents need to cooperate effectively by using partial knowledge about each other. The impact of proactive information gathering on the evolution of cooperation is explored through simulation experiments, and the results will be explained in Chapter 5.

The design of this new model is expected to allow the agents to recognize cooperative and non-cooperative behaviours and adjust their behaviour accordingly. Therefore, deciding how advanced agents will use the gathered information is an important question to answer. More particularly, we are interested in investigating the impact of agents' different strategies on the evolution of cooperative behaviours.

3.2 The IG-Model

To answer the proposed research question, we need to compare the agents' performance with and without information gathering (IG) technique. For that purpose, we adopt the premise of the basic model. In the basic model, agents always play their *native* strategy which is either cooperation or defection, therefore we call them *naive* agents. In the new model, named the *IG-model*, we add our desired features by introducing *advanced* agents. That modification will result in having a combined population of four types of agents:

- 1. naive-cooperators do not collect information and always play cooperation as their native strategy.
- naive-defectors do not collect information and always play defection as their native strategy.
- 3. advanced-cooperators can selectively gather information about other agents, and their goal is developing cooperative behaviours.
- advanced-defectors can selectively gather information about other agents. They try to play with cooperators to maximize their utility which leads to reproduction.

3.2.1 Advanced Agent

Advanced agents' information gathering involves two different parts. First, advanced agents can retain a personal game memory. When an advanced agent plays with another agent, it keeps a record of the identity of that agent and its actions. Also, it can acquire information about the past behaviour of its opponent, which is saved as the *history*. The available information in each agent's history includes its action in previous games, the actions of players with whom it played, and identity of those players. After gathering the information, advanced agents adjust their strategy based on the collected information, explained in the following section.

3.2.2 Strategy Adjustment

Each advanced agent has a short-term and a long-term agenda, and it needs to move towards the defined agenda in choosing its strategy. An advanced cooperator avoids interacting with defectors and tries to develop cooperation, even if that action reduces its short-term benefit. In particular, an advanced cooperator can defect when it plays with a naive cooperator, knowing that a naive cooperator always cooperates, and defection has the highest payoff. However, the long-term agenda for an advanced cooperator is to help other cooperators to build sustainable clusters, which eventually leads to the evolution of cooperation.

On the other hand, an advanced defector tries to guarantee its survival and reproduction; then it needs to interact with cooperators. However, the advanced defector does not need too many cooperators for its survival and reproduction. So, it is true that an advanced defector prefers to maximize its payoff in the short run, but it can have a long-term agenda of being part of a small cluster of cooperators and not allowing them to grow in size.

Each agent plays its native strategy in its very first game by default. So, if an advanced agent gathers information about its opponent's first game, it can detect the opponent's identity as a cooperator or a defector. Then, the advanced agent can select its strategy based on the gathered information. An advanced cooperator will play cooperation when its opponent cooperated in the first game, and it will play defection when its opponent defected in the first game. In the absence of such information, an advanced cooperator will cooperate.

An advanced defector should engage in some measured cooperation with cooperators in order to keep enough of them alive for itself to survive. For that purpose, an advanced defector should be aware of the frequency of its encounters with cooperators (as determined by checking their first game). Therefore, to avoid losing too many cooperators, an advanced defector begins cooperating with them when the frequency of its encounters with cooperators falls below a certain threshold. When the frequency of encounters with cooperators rises above a predefined threshold, an advanced defector resumes pure defection. In the absence of any information about the opponent's first game, an advanced defector will defect. The specified threshold is determined by *frequency threshold* \pm *deviation*. The values of frequency threshold and deviation are calculated experimentally and explained in Chapter 5. In the absence of any information, an advanced defector will defect. Also, an advanced defector always plays defection with another defector.

As discussed in Section 2.6, different models have been introduced to formalize the evolution of cooperation among agents through direct or indirect reciprocity with the use of trust, reputation, and social information [Mohtashemi and Mui, 2003] [Sabater and Sierra, 2002] [Sekaran and Sen, 1995]. In those models, information gathering mostly happens through observation, direct interaction, interviewing others, or following social networks [Sabater and Sierra, 2002]. In our model, an agent does not necessarily need to establish a direct interaction to collect information, and it only gathers factual historical information. Then, an agent can reason about its strategy and adjust its strategy based on the information it receives about the behaviour of other agents. There are also some theoretical models in which cooperation can evolve by equilibriums. However, our model differs from those models in few key aspects. Unlike the model described in [Takahashi, 2010], the agents only gather information about the opponent's first game in our model. Also, players can send a message to each other, prior to the game, in [Awaya, 2014] and [Bachi et al., 2010] models. However, there is no communication between the agents in our model.

3.3 Cluster studies and neighbourhood profile

To have a better understanding of if and how cooperation evolves, one could investigate the change in the number of agents in the spatial neighbourhoods and the individual clusters. Smaldino et al. [2013] refer to a group of cooperators which can manage to stay alive with repeated mutual cooperation as a cluster of cooperators. The clusters are usually surrounded by defectors on the periphery. Those defectors which are not closed to a cluster of cooperators will die, since they cannot interact with cooperative partners.

Measuring the composition of individual clusters would help to understand the property of clusters, which shrink in size or prosper and grow fast. To identify all the agents in a cluster explicitly, the Tarjan's depth-first search (DFS) algorithm [Tarjan, 1972] has been used.

In addition, evaluating the spatial environment of an agent of a given type in its spatial neighbourhood would help us to identify if and how the number of agents from one specific type is changing in another agent type's neighbourhood.

Chapter 4

Simulator Architecture

This chapter presents the simulator architecture. It begins by identifying the simulator requirements in Section 4.1. Section 4.2 describes the simulator structure. Section 4.3 elaborates upon the behavioural view of the simulator's different components. Section 4.4 demonstrates our approach to the experimentation.

4.1 The Simulator Requirements

The list of requirements for the simulator is divided in three categories: *functional* requirements, non-functional requirements and domain-specific requirements [Sommerville, 2011]. In the rest of this section, different components of each category are described.

4.1.1 Functional Requirements

The functional requirements of the system are captured in the use case diagram represented in Figure 4.1. The use case diagram specifies different interactions between the simulator and the experimenters. The experimenter runs the simulator to initialize the experiments and receives the results. Each use case is described in more detail in the following sections.

4.1.1.1 Set and Run Experiment

The experimenter can set an experiment and run it. This use case includes other use cases as follows:

- 1. *Configure Experiment*: The experimenter can configure a new experiment with respect to the desired models and parameters.
 - (a) Set Parameters: The experimenter should initialize the environment, agents, game, and simulator parameters.
 - i. *Environment Parameters*: The properties of the environment including the grid size and the environmental cost of living.
 - ii. Agents Parameters: The experimenter can set the parameters for the agents. The agent parameters include agents' initial and maximum allowed level of energy. In addition, the experimenter should indicate if agents receive a fixed or a random initial energy value. The experimenter can also configure the size of personal game memory and the discipline for checking the opponent's history for advanced agents.



Figure 4.1: The use case diagram of the simulator

iii. Game Parameters: The experimenter can set payoff values, which must

satisfy the constraints defined in Section 2.2.

- iv. *Simulator Parameters*: The experimenter can configure different parameters of the simulator including the initial number of agents and the maximum carrying capacity.
- (b) Set the Model: The experimenter can run different models, particularly the basic model and the IG-model.
- 2. Save Experiment: The experimenter can save the configuration for an experiment.
- 3. Load Experiment: The experimenter can load an experiment that has been saved previously.
- 4. *Run Experiment*: The experimenter can control the execution of an experiment. The experimenter can start and stop an experiment. Also, the experimenter can run an experiment, step-by-step, to monitor the course of the experiment. It is worth noting that an experiment would stop if the defined maximum number of rounds is reached or if all agents die.

4.1.1.2 Display Experiment

- 1. Display Experiment Course: The experimenter can invoke the simulator to display a graphical representation of an experiment; this includes displaying the environment as a square grid, and the agents with their movement as well as their reproduction or death. In addition, the experimenter can select to observe the behaviour of individual clusters.
- 2. *Display Results*: Once the experimenter enables a graphical representation of a simulator, the simulator shows a numerical and a graphical representation of

the final results. The final results indicate the total number of agents of each type that remain alive at the end of an experiment.

3. *Debug Results*: The experimenter can select to view the logs, which are generated by the simulator. The information in the log window describes the course of an experiment in greater detail and is used to study and analyze the results.

4.1.2 Non-functional Requirements

- The simulator needs to provide the support for the processing and analyzing of simulation results. For that purpose, the simulator interoperates with MySQLTM database [MySQL[®], 2017] to store the collected results from the experiments. The collected data can be viewed using the MySQLTM Workbench [Inan and Juita, 2011]. MySQLTM Workbench is the official integrated environment for MySQLTM to manage the database design and modeling.
- 2. The simulator should provide a graphic user interface (GUI) to have interactive experimentation. The GUI should represent the functional requirements of the simulator.

4.1.3 Domain-specific Requirements

The domain-specific requirements specify the details of simulator structure, with respect to the models it simulates, and reflect the fundamental concepts of the multiagent system (MAS) and agents interaction. The simulator has been developed to study the evolution of cooperation among self-interested agents with and without proactive information gathering. The models of multiagent systems studied by using the simulator are: the basic model, introduced by Smaldino et al. [2013], and the IG-model (described in Section 3.2). The domain-specific requirements for the above models are as follows:

- 1. The simulator should simulate MAS, which contains a number of agents, situated in an environment.
- 2. The simulator should represent an energy-based life cycle for the agents.
- 3. The simulator should support the presence of different types of agents, namely the *naive* agents and the *advanced* agents.
- 4. Individual agents need to perform various actions, which includes interacting with each other and moving in the environment. Also, individual agents need to be able reproduce new agents of their own type.
- 5. The agents interactions should follow the Prisoner's Dilemma (PD) game structure.
- 6. Advanced agents should be able to adjust their strategy based on the gathered information, in order to study the impact of information gathering upon the evolution of cooperation,
- 7. The environment should present a source of adversity, called *harsh environmen*tal conditions.
- 8. The environment should be toroidal, which resembles a torus [Oxford University Press, 2017].

It should be noted that, the environment can present a source of uncertainty, in order to represent the limitations on the information acquisition.

4.2 The Simulator Structure

To support the described requirements, our simulator contains three main components: *controller*, *MAS-layout*, and *front end*. The high-level structure of the simulator is shown in Figure 4.2.



Figure 4.2: The high level structure of the simulator

The controller is responsible for running and controlling the experiments. The MAS-layout represents the model for the simulation. Each model includes agents, an environment in which agents are located, and the interaction between the agents. The front end is a user interface which allows the experimenter to setup and control the course of the experiment and presents the results. Each component is further comprised of other components. In the rest of this section, each component is explained in greater detail.

4.2.1 Controller

The controller is the main component of the simulator. The functional components of the controller coordinate with each other to initialize and execute the experiments and collect the results. Figure 4.3 shows the internal structure of the controller.



Figure 4.3: The controller

Each experiment involves a number of agents—divided into different types—which are acting on the same environment. Agents can attempt to play the PD game, move in the game space, and reproduce new agents. The design of the controller's components supports agents' actions and allows an experiment to proceed in discrete rounds.

Executing an experiment begins with parameter initialization. The *parameters* manager is designed to manage the parameters' configuration for the simulator. The experimenter sets the initial parameters through the front end. The parameters manager receives the parameters, validates the values, and passes them to the *experiment* handler.

The experiment handler is responsible for executing an experiment. An experiment *run* refers to different instances of the same experiment, with the same parameter settings and independent randomization of random parameters under the same distributions. The experiment handler interacts with the parameters manager to receive the initial parameters setting and activate the runs.

The *run handler* executes each run of an experiment and presents the final results when one run has ended. The final results indicate the total number of the agents of each type that have survived at the end of an experiment.

Each run of an experiment consists of a specified number of rounds, and it will stop if it reaches the maximum number of rounds. The *round handler* manages the execution of each round in a run of an experiment. In each round, an agent can be interacting with other agents by playing the PD game or by moving in the environment. The *game handler* executes the PD game between two agents and modifies each agent's energy level based on the game payoffs. For the advanced agents, the game involves the information acquisition and strategic decision making, explained in the Section 4.2.2.

Finally, there is a service component called *logger*, which works with the run handler to collect all generated logs in the simulator. The logs show the progress of an experiment in different rounds for different agents and can be used to debug the experiment's results.

4.2.2 MAS-layout

The MAS-layout represents a multiagent system. This component represents an evolving population of agents with an energy-based life cycle, situated in the environment. The agents are interacting with each other, moving in the game space, and reproducing if applicable. The MAS-layout's components are described in the following:

• Information Repository

The MAS-layout contains the *information repository* which keeps records of all the agents' interactions. The identity of agents, which played with each other, and their actions in each game are stored in the information repository. The advanced agents can interact with the information repository to acquire information about their opponents.

• Agent

The simulator provides a simple architecture for agents to support different models of population and agents' interactions. All agents are capable of playing the game, moving, and reproducing. In addition, advanced agents are able to acquire information from the information repository.

The advanced agent architecture consists of a personal game memory, a memory updater, and a strategic reasoning engine (Figure 4.4). There is a corresponding record in an advanced agent's personal game memory for each game it plays. Each record in an advanced agent's memory includes the identity of one particular opponent and its action in each game. The capacity of the memory can be limited to a specific number, or it can be infinite.

Each advanced agent acquires information about its opponents by checking their history through the information repository. Once an advanced agent has some new information about other agents in the population, it updates its personal game memory through the memory updater. It should be mentioned that the simulator is designed with the ability of presenting some limitations on information acquisition process. That feature can be used for the future works.

The advanced agent then uses the information it has about its opponent to decide its strategy. The *strategic reasoning engine* allows the advanced agents to reason about their opponent, based on the information they have, and make decisions as discussed in Section 3.2.2.

• Environment

The environment provides the game space for agents to play, move, and reproduce. All agents' interactions, movement, and reproduction happen on a toroidal square grid adopted from the basic model. The main component of the environment is the *board*. The board is a square grid with periodic boundaries. The agents are situated in unique cells on the board.



Figure 4.4: The advanced agent architecture

4.2.3 Front End

The front end provides the experimenter's access to the simulator. It allows the experimenter to observe and control the course of the experiment. The front end consists of a control panel and a graphical user interface (GUI). The internal structure of the front end and its connection to other components of the simulator are shown in the Figure 4.5.

The experimenter can access the simulator and control an experiment through the control panel. The experimenter can initialize, run, save, and load an experiment by using the control panel.

The experimenter can observe the course of simulation through the GUI. By observing the experiment, the experimenter can follow the change in the population in general or in a specific round by stepping through an experiment. The GUI also shows the final results, which is the number of agents of each type that survived at the end of an experiment.



Figure 4.5: The front end

4.3 The System Behaviour

This section explains the behavioural aspects of the simulator's components. The experimenter adjusts the parameters to generate an experiment through the control panel of the front end. The control panel window is shown in the Figure 4.6.

🛃 Parameter Setup	_ _ x
<u>File View</u>	
GENERAL SETTING GUI OPT SEED: 406290462022 Random SEED Board Size: 100 Capacity: 50 Initial Number of Agents: 40	TION PARA METERS
Number of Iterations: 2000 Number of Experiments: 1 + + Active Th AGENT SETTING PAYOFF MATRIX T: 5 8: 8: 0 9: 0 Maximum Initial Energy: 50 Image: Fixed Energy S: 0 9: 0 9: 0 Reproduction Energy: 50 Image: Fixed Energy S: 0 9: 0 9: 0 Birth Threshold: 100 Image: Simple S	reads: 1 ORS
ENHANCED PROTOCOL SETUP Cooperation Portion: 10 Defection Portion: 10 Naive Defection Portion: 10 Memory Depth: Infinite First Query Probability: 100 Mixed Probability: 100 Mixed Probability: 100 Mixed Probability: 100 Mixed Probability: 100 Mixed Probability: 100 Mixed Probability: Infinite Mixed Probability: Infinite Infinite Mixed Probability: Infinite Mixed Probability: Infinite Mixed Probability: Infinity	ORDER • Last %
Generate	

Figure 4.6: The control panel

The experiment handler receives the initial parameters from the parameters manager at the beginning of each experiment, and it activates the run handler for executing different runs of the same experimental settings (Figure 4.7). The number of runs must be adequately large to assure the statistical significance of the results.



Figure 4.7: The mechanics of the *experiment*

The run handler first initializes the environment and the population of agents for a new experiment run. Based on the specified model, the population can include four different types of agents: Advanced-Cooperator, Advanced-Defector, Naive-Cooperator, and Naive-Defector. Agents are randomly located in separate cells for each run of an experiment. Also, agents are assigned with an initial energy value. This value is either assigned randomly by the uniform distribution (U[a, b], where a is the minimum and b is the maximum) or constantly. Each run of an experiment consists of a number of rounds, specified by the experimenter. The run handler activities the round handler for executing each round (Figure 4.8).



Figure 4.8: The mechanics of the run

At each round, every agent, that has not already played the game that round, searches its neighbourhood for an opponent that has also not already played that round. Note that each agent's neighbourhood consists of eight closing cells. If the agent finds an opponent, they play the PD game and receive payoffs. The received game payoffs change the energy level of each agent. However, agents' energy level is always capped at a constant amount. Therefore, an agent cannot accumulate energy without bound.

If there is no opponent, the agent searches its neighbourhood to move to an empty

random cell. If an agent moves to another cell, it should wait for the next round to look for an opponent again. Thus, the agent is not able to move and play in the same round. If there is no empty cell in an agent's neighbourhood, the agent stays in its current cell and waits for the next round (Figure 4.9).



Figure 4.9: The mechanics of the round

The game handler controls the course of playing the game between two agents. Naive agents play their native strategy in any instance of a game. The native strategy of naive cooperators and naive defectors is cooperation and defection respectively. Advanced agents are expected to reason about their strategy based on the information they have obtained about the behaviour of other agents. The advanced agents are designed to play different strategies in order to achieve their short-term and long-term goals as explained in Section 3.2.2.

At the end of each round, the cost of living is deducted from each agent's energy level, and the round handler checks each agent's energy level. If an agent's energy meets the birth threshold, that agent is ready to reproduce a new agent. The reproduction is successful if there is an empty cell in the agent's neighbourhood and the population maximum carrying capacity is not reached. Agents yield a predefined energy value to their child in the case of reproduction. A newly born agent is the same type of its parent. On the other hand, if an agent's energy passes the death threshold, that agent is removed from the game space. Figure 4.10 shows the activity diagram for agents during one round.



Figure 4.10: The agent's activity diagram

During a round agents are in different states. The followings are the most important states which determine the agents' behaviour based on their interaction and movement. The Figure 4.11 illustrates the state-machine representation for an agent during a round.

- *Prisoner*: the agent has found an opponent. In this state, other agents cannot select a prisoner agent as an opponent anymore.
- *Moved*: the agent has not found an opponent and, it has moved to another cell. Since an agent which moves cannot play the game in the same round, other agents are not able to select a moved agent as an opponent anymore.
- *Ready-to-Reproduce*: the agent's energy level has reached to the birth threshold, and it is ready to reproduce a new agent. If the ready-to-reproduce agent does not give birth to new agent, its state will be changed to the ready for the next round.
- *Ready*: the agent is ready to find an opponent for playing the game or move to another cell. Ready is a final state in a round, which means when agent is ready it should wait for the next round to begin.
- *Dead*: the agent's energy level has passed the death threshold, and it should be removed from the game space.



Figure 4.11: Double-line boundaries indicate final states for an agent during one round. The final states are shown more than once for the purpose of better presentation.

When one round is over, the round handler modifies the population size based on the number of births and deaths. It saves the corresponding information in a database and passes that information to the run handler. The run handler checks if the run is over. In the case that all agents die or the maximum number of rounds is reached, one run is considered as finished.

If the experimenter invokes the GUI, the experiment handler presents the final results through the GUI at the end of each run. By invoking the GUI, the experimenter can also select to watch the course of simulation at one specific round. Figures 4.12 and 4.13 show an example of the course of simulation in the GUI window. In addition, the simulator has a mouseover tool. The user can gather more information about one agent by moving the mouse over that particular agent. The agent's ID, its strategy, and its energy appear on a small box. Figure 4.14 presents an example of mouseover tool.

Also, one of the fundamental goals of designing the GUI is to visualize the collected data in the simulator. For that purpose, we design the *report viewer*, which is a visualization tool and provides the support for drawing charts. The accumulated data is displayed by various graphs using the report viewer. Figure 4.15 shows an example of a chart created by the report viewer.

To study the individual clusters, one more tool has been added to the simulator, named the *cluster viewer*. The cluster viewer is used to identify the clusters in the course of a simulation. The experimenter can use the saved experimental results to run the cluster viewer. The cluster viewer is also empowered with some additional features such as moving back and forth between different rounds of the game, filtering clusters based on their size, and filtering agents based on their energy level. Figure 4.16 shows an example of cluster in the cluster viewer.



Figure 4.12: The blue and green dots are cooperators, and the red and purple dots are defectors



Figure 4.13: The GUI window, when we only have two types of agents in the population

🔬 Cluster Viewer	er CLUSTERS MINIMUM SIZE FILTER								- 0 ×			
Experiment Number	897627652654 - AD 0.5	•	1	11 21	31 41	51 61	71 81	91 101		Round Previous	0 💌 🅪 Next	
Apply Energy Filter		1	2	3	4	5	6	7	8	9	10	
	1											
	2	D	С	D	С							
Live Pool	3			D)	COOPERA	TION .						
Cooperator: 4 Naive Cooperator: 4 Defector: 4 Naive Defector: 4	4		С		Energy: 50	0.00				D		
	5											
	6					C						
	7	D					D)			D		
Experiment Specs	8		C		С							
Birth Threshold: 100 Capacity: 20	9		D	C								
P: 0.00 R: 3.00 S: -2.50 T: 5.00 Cost Of Living: 0.50	10								C			

Figure 4.14: The Mouseover tool



Figure 4.15: The report viewer

💰 Cluster Viewer					CLUSTERS	MINIMUM SIZE FILTER				-	o ×
Experiment Number	897627652654 - AD 0.5	•	1	11 21	31 41	51 61	71 81	91 101		Round Previous	▼ Mext
Apply Energy Filter		1	2	3	4	5	6	7	8	9	10
	2		С	R	С						
Live Pool Cooperator: 4 Naive Cooperator: 4 Defector: 4 Naive Defector: 4	3 4 5	(С	D						D	
Experiment Specs Board Size: 10 Agent Max Energy: 151 Birth Threshold: 100 Capacity: 20 P: 0.00 R: 3.00 S: 2.50	6 7 8 9	D	C	С	С	С	D		C	D	
T: 5.00 Cost Of Living: 0.50	10	(0.4) (0									
Cooperator: 2 Naive Cooperator: 1 Defector: 1 Naive Defector: 1	Centeral Agent: 5/Locatio	on: (1, 1)/ strategy: DEFE	711ON **								

Figure 4.16: The Cluster and its information

4.4 Experimentation

In the different series of experiments, we explore if and how the total number of agents from different types can be influenced by the proactive information gathering. The simulation models are based on the same parameter settings that were used in the basic model. However, we might need to change some parameters' values to find the optimized situations in which cooperation evolves.

In a preliminary round of experiments, we enable only one type of advanced agents, either cooperators or defectors, to figure out how they behave in different conditions. The increase or decrees of the number of agents depend on different factors, such as the cost of living, the birth threshold, and the population maximum capacity. Running experiments with a single type of advanced agents give us a better understanding of their behaviour under different conditions.

Once we we optimize the values for different parameters, In the next round of our simulation experiments, we explore the impact of proactive information gathering upon the behaviour of agents in the presence of both advanced cooperators and advanced defectors.

Chapter 5

Evaluation

This chapter presents the simulation results for different models that employ the proactive information gathering technique, in comparison with the basic model. We first examine our simulator by running the basic model. Then, we compare the enhanced model through a series of experiments for different values of parameters representing environmental adversity, and agents' energy level.

5.1 Parameter Space

- board-size: the environment includes a 100×100 discrete square grid with periodic (toroidal) boundaries.
- agent-initial-energy: each agent receives the fixed starting energy of 50. In the basic model of [Smaldino et al., 2013], agents are initialized by a random energy value with the uniform distribution. To eliminate the effect of random variables,

we initialized all the agents with the fixed value of the starting energy. It should be mentioned that we repeated all the experiments from the basic model with the fixed starting energy value for the sake of comparison.

- agent-max-energy: the maximum allowed energy level for an agent is 150.
- birth-threshold: the energy threshold required for reproduction is 100.
- reproduction-energy: each time an agent reproduces, it yields 50 units of its energy to its child.
- max-population: the maximum allowed population size is 5000 unless noted otherwise.
- payoff: The payoffs are set as follows: T = 5, R = 3, and P = 0.
- environmental adversity: the payoff value of unreciprocated cooperation, S, and the value of the cost of living vary in different experiments.
- frequency threshold: the value of frequency threshold varies in different experiments.
- deviation: the value of the deviation is set to 0.1.
- number of runs: in order to get statistically significant results, each experimental setting has been examined for 50 runs. The number of agents is averaged over 50 simulation runs. It should be mentioned that each run of an experiment may produce different results due to some random factor, including the agents' initial composition, their movement, and partner selection.
- number of rounds: each run of an experiment includes 10000 rounds.
5.2 The Basic Model

Smaldino et al. [2013] examine the impact of two factors related to environmental harshness. Under certain assumptions of the basic model, Smaldino et al. [2013] conclude the following results:

When the cost of living has a fixed value, and the cost of unreciprocated cooperation (corresponds to the payoff S) varies, the more punishing value for the cost of unreciprocated cooperation results in losses for cooperators at the beginning; however cooperative agents can survive in the long run. By decreasing the cost of unreciprocated cooperation, it is more difficult for cooperators to keep expanding cooperative clusters. Figure 5.1 shows the number of cooperators regarding the different values of S in our simulator, which closely resembles those from [Smaldino et al., 2013]. Also, Figure 5.2 presents the number of defectors with respect to the different values of S.



Figure 5.1: The number of cooperators for different values of S when cost of living is fixed at 0.5. The graph presents the averaged results over 50 simulation runs.



Figure 5.2: The number of defectors for different values of S when cost of living is fixed at 0.5. The graph presents the averaged results over 50 simulation runs.

Observing the population development in the basic model, when cost of living is constant and S varies, shows that there are some oscillations in the number of agents as we can see in Figures 5.1 and 5.2. One should notice that increasing the population capacity or running the desired experiments more than 10000 rounds, can result in a more stable situation for both cooperators and defectors as mentioned in [Smaldino et al., 2013]. Figure 5.3 presents an example of population development with respect to the number of cooperators and defectors, when population capacity has a higher value and experimental results are gathered beyond 10000 rounds.



Figure 5.3: The number of defectors and cooperators in the basic model when S is -1, cost of living is 0.5, population capacity is 9000, and number of rounds is 30000. The graph presents the averaged results over 50 simulation runs.

2. When the cost of unreciprocated cooperation is constant, and the environmental cost of living takes different values, the higher cost helps cooperators in the long run unless the cost of living is too high for any individual to survive. Figure 5.4 shows the number of cooperators regarding the different values of cost of living. Also, Figure 5.5 presents the number of defectors with respect to the different values of cost of living. It should be mentioned that the cost of living is reduced from all agents' energy and affect both cooperators and defectors in the same way. Where S is fixed at 0, cooperators are not punished for interacting with defectors. Therefore, the population reached its maximum capacity and stabilized sooner unlike when S takes more punishing values.



Figure 5.4: The number of cooperators for different values of cost of living when S is fixed at 0. The graph presents the averaged results over 50 simulation runs.



Figure 5.5: The number of defectors for different values of cost of living when S is fixed at 0. The graph presents the averaged results over 50 simulation runs.

5.2.1 Clusters and Neighbourhood Profiles

As we can see in Figures 5.1 and 5.4, there is a drop in the number of cooperators at the beginning. Decreasing the number of cooperators eventually leads to the death of defectors, since defectors need to have the efficient amount of interactions with the cooperators to survive. However, if some cooperators can find each other and stay alive by repeated mutual cooperation, they can develop a cooperative cluster. Those defectors, which are closed to a cooperative cluster, can survive. To have a better understanding of the described phenomena, we have developed two extra tools to study the behaviour of agents more precisely. Using those tools we study the neighbourhood profile of each agent type and individual clusters.

In the study of individual clusters, we have identified two different cluster types. One cluster is the *rising* cluster, in which the number of agents increases and cluster grows. The other type of cluster is called the *dying* cluster, in which the number of agents decreases, and all the agents in that cluster die.

Figures 5.6a, 5.6b, and 5.6c present an example of a dying cluster in the basic model.



Figure 5.6: An example of a dying cluster in the basic model when cost of living is 1 and S is 0. Cooperators are gray and defectors are yellow.

Figures 5.7a, 5.7b, and 5.7c present an example of a rising cluster in the basic model. Although the number of cooperators is less than the number of defectors on that cluster, it forms a pattern in which a small cluster of cooperators is surrounded by defectors. That pattern stays relatively stable over different rounds.



Figure 5.7: An example of a rising cluster in the basic model when cost of living is 1 and S is 0. Cooperators are gray and defectors are yellow.

The observation of local neighbourhood profiles in individual clusters shows that:

- 1. In a dying cluster, the number of defectors in a cooperator's neighbourhood is higher than the number of cooperators (Figures 5.8a and 5.8b).
- 2. In a rising cluster, the number of cooperators in a cooperator's neighbourhood is higher than the number of defectors (Figures 5.9a and 5.9b).



(a) Number of Agents



Figure 5.8: An example of the number of agents and cooperators' neighbourhood profile in a dying cluster captured in Figure 5.6



Figure 5.9: An example of the number of agents and cooperators' neighbourhood profile in a rising cluster captured in Figure 5.7

5.3 The IG-Models

This section presents the results from the IG-models. There are three different sets of experiments in which we investigate the impact of information gathering (IG) upon the evolution of cooperation. In the first series of experiments, we have enabled only the cooperators to gather information, IG-AC model. In the second series of experiments, we only have the advanced defectors, IG-AD model. In the last series of experiments, we have enabled both defectors and cooperators to gather information, IG-ACD model.

5.3.1 Initial Agent Population

The initial agent population can contain all the agent types or some of them based on the model. The initial number of agents in all the experiments is 1600 and is equally divided into the different types of agents. However, the actual population composition depends on the model.

- In the basic model, the initial population includes 800 naive cooperators and 800 naive defectors.
- In the IG-AC model, the initial population includes 400 naive cooperators,
 400 advanced cooperators, and 800 naive defectors.
- In the IG-AD model, the initial population includes 800 naive cooperators,
 400 advanced defectors, and 400 naive defectors.
- In the IG-ACD model, the initial population includes 400 naive cooperators, 400 advanced cooperators, 400 naive defectors, and 400 advanced defectors.

5.3.2 IG-AC

To have a simple model to start, where we can interpret the result easier, we decided to set the advanced agents' strategies as follows:

- The advanced cooperator A (after the first game) checks the first game that its next opponent B has played. If there is none, A cooperates. Otherwise, it now knows whether B is a cooperator or a defector (because it knows B's native strategy) and plays cooperation or defection accordingly.
- All defectors are naive.

5.3.2.1 Observations and Analysis

- 1. Comparing the number of cooperators in the basic and IG-AC models for the different values of S (Figures 5.1 and 5.10) shows that cooperators have a better performance in the IG-AC model than the basic model.
- 2. Comparing the basic and IG-AC models for the different values of both S (Figures 5.1 and 5.10) and cost of living (Figures 5.4 and 5.11) shows that the number of cooperators which die at the beginning of an experiment in the basic model is higher than the number of cooperators which die at the beginning of an experiment in the IG-AC model. So, it is more difficult for cooperators to survive and reproduce without the information gathering technique.
- 3. When the population reaches its maximum allowed capacity, we can see some stability and fewer changes in the number of agents. The point of stability is reached sooner in the IG-AC model than the basic model.



Figure 5.10: The number of cooperators (naive and advanced) for different values of S when cost of living is fixed at 0.5 in IG-AC model. The graph presents the averaged results over 50 simulation runs.



Figure 5.11: The number of cooperators (naive and advanced) for different values of cost of living when S is fixed at 0 in IG-AC model. The graph presents the averaged results over 50 simulation runs.

5.3.2.2 Clusters and Neighbourhood Profiles

Studying neighbourhood profiles of agents in the IG-AC model show that on the one hand, the number of cooperators in both cooperators and defectors neighbourhood increases. On the other hand, the number of defectors in both cooperators and defectors neighbourhood decreases. Figure 5.12 shows an example of how the number of neighbours in each agent type neighbourhood changes over different rounds.



Figure 5.12: The number of neighbours in different types of agent's neighbourhood when cost of living is 0.5 and S is -2.5 in IG-AC model. The graph presents the averaged results over 50 simulation runs. The graph presents the averaged results over 50 simulation runs.

Like the basic model, there is a drop in the number of (naive) cooperators at the beginning of the IG-AC model as well. However, the number of advanced cooperators is increasing, which results in the higher number of cooperators in total. Due to the difference between the number of different types of cooperators, we will see some scattered clusters when an experiment starts. However, advanced cooperators grow fast in size. As a result, the clusters in which advanced cooperators are dominant, form really quick (Figures 5.13a, 5.13b, and 5.13c).



Figure 5.13: Clusters' development in the IG-AC model when cost of living is 0.5 and S is -2.25. Naive cooperators are gray and naive defectors are yellow. Advanced cooperators are blue and advanced defectors are red.

5.3.3 IG-AD

Then, we decided to keep cooperators naive and introduce advanced defectors. As mentioned before, defectors need to interact with cooperators to gain enough energy for survival and reproduction. For this purpose, an advanced defector should be aware of the frequency of its encounters with cooperators (as determined by checking their first game). Therefore, to avoid losing too many cooperators, an advanced defector begins cooperating with them when the frequency of its encounters with cooperators falls below a certain threshold. When the frequency of encounters with cooperators rises above a predefined threshold, an advanced defector resumes pure defection. In the absence of any information about the opponent's first game, an advanced defector will defect.

The value of frequency threshold and deviation are calculated based on some experimental results. In the basic model, defectors are most successful when S is 0 and cost of living is 0.5. Under this condition, the frequency of their interaction with (naive) cooperators for (naive) defectors is 0.3. So, we used 0.3 as the best value for the frequency threshold, f. Running more experiments of the basic model showed that defectors are still successful when the initial number of cooperators is 0.2 of the initial number of defectors, but not when the initial number of cooperators is 0.15 of the initial number of defectors. Therefore, to ensure the stability, we use 0.1 as the best value for deviation, d. It should be noted that when frequency threshold has a higher value, defectors would cooperate with cooperators most of the time. That behaviour is not expected from a defector. However, we examined the other values for frequency threshold such as 0.5 and 0.7 to see how that can impact the general population development.

5.3.3.1 Observations and Analysis

• Observations

- **O.1** When the cost of living is fixed at 0.5, the number of cooperators decreases by decreasing the value of S (Figures 5.14, 5.16, and 5.18) which does not happen in the basic model.
- O.2 Increasing the cost of living has the same impact as we observe in the basic model and leads to the evolution of cooperation (Figures 5.15, 5.17, and 5.19).



Figure 5.14: The number of cooperators for different values of S in the IG-AD model when cost of living is fixed at 0.5 and frequency threshold is 0.3. The graph presents the averaged results over 50 simulation runs.



Figure 5.15: The number of cooperators for different values of cost of living in the IG-AD model when S is fixed at 0 and frequency threshold is 0.3. The graph presents the averaged results over 50 simulation runs.



Figure 5.16: The number of cooperators for different values of S in the IG-AD model when cost of living is fixed at 0.5 and frequency threshold is 0.5. The graph presents the averaged results over 50 simulation runs.



Figure 5.17: The number of cooperators for different values of cost of living in the IG-AD model when S is fixed at 0 and frequency threshold is 0.5. The graph presents the averaged results over 50 simulation runs.



Figure 5.18: The number of cooperators for different values of S in the IG-AD model when cost of living is fixed at 0.5 and frequency threshold is 0.7. The graph presents the averaged results over 50 simulation runs.



Figure 5.19: The number of cooperators for different values of cost of living in the IG-AD model when S is fixed at 0 and frequency threshold is 0.7. The graph presents the averaged results over 50 simulation runs.

O.3 Comparing the basic model with the IG-AD model both defectors and cooperators are doing better in the IG-AD model than in the basic model when the frequency threshold has a higher value. Also, both defectors and cooperators are doing better in the IG-AD model than in the basic model, or the same as in the basic model when the frequency threshold has a lower value. Figures 5.20 and 5.21 show some examples of a comparison between the average number of agents in the basic model and in the IG-AD model.



Figure 5.20: The number of cooperators in the basic and the IG-AD models when S is 0, cost of living is 0.5 and frequency threshold varies. The graph presents the averaged results over 50 simulation runs.



Figure 5.21: The number of defectors in the basic and the IG-AD models when S is -2.5, cost of living is 1 and frequency threshold varies. The graph presents the averaged results over 50 simulation runs.

- **O.4** In the case of low cost of living, when the value of unreciprocated cooperation is zero, the number of naive defectors is higher than the number of advanced defectors. This difference is increasing when frequency threshold has a higher value (Figures 5.22a, 5.22b, and 5.22c).
- **O.5** In the case of low cost of living, when unreciprocated cooperation has a more punishing value of -2.5, the number of naive defectors is lower than the number of advanced defectors. In this case, when the frequency threshold is low (0.3), even in some experiments, the actual number of naive defectors is higher than the advanced defectors. However, the difference between the number of advanced defectors and naive defectors increases by increasing the frequency threshold (Figures 5.23a, 5.23b, and 5.23c).



Figure 5.22: The number of advanced defectors and naive defectors in the IG-AD models when S is 0, cost of living is 0.5, and frequency threshold takes different values. The graph presents the averaged results over 50 simulation runs.



Figure 5.23: The number of advanced defectors and naive defectors in the IG-AD models when S is -2.5, cost of living is 0.5, and frequency threshold takes different values. The graph presents the averaged results over 50 simulation runs.

• Analysis

A.1 Cost of unreciprocated Cooperation

Under the environmental harshness conditions represented by the cost of unreciprocated cooperation, decreasing the value of S slows down the evolution of cooperation unlike the basic model (observation **O.1**). It should be noted that decreasing the value of S has a more direct impact on cooperators.

A.2 Cost of Living

Under the environmental harshness conditions represented by the cost of living, increasing the cost of living results in the evolution of cooperation (observation **O.2**). However, when the cost of living has a lower value, defectors, in general, are doing better than the cooperators. It worth to mention that cost of living effects both cooperators and defectors in the same way.

A.3 Environmental Harshness

Based on the observation **O.3**, under the lower environmental harshness conditions represented by both unreciprocated cooperation and environmental cost of living, cooperators have a better performance in the IG-AD model than in the basic model. Also, under the higher environmental harshness conditions represented by both unreciprocated cooperation and environmental cost of living, which is in favour of cooperators, the number of defectors in the IG-AD model is higher than the number of defectors in the basic model. Therefore, enabling the advanced defectors to base their strategy on the gathered information:

- (a) helps defectors to grow more in size when the environment is too harsh, which does not happen in the basic model.
- (b) allows cooperators to sustain cooperation under the lower threshold of environmental harshness.
- A.4 Number of Defectors

As mentioned in observation **O.5**, when the cost of living is low and unreciprocated cooperation has a more punishing value of -2.5, the number of naive defectors is lower than the number of advanced defectors. Also, the difference between the number of advanced defectors and naive defectors increases by increasing the frequency threshold. In this situation, cooperators start to disappear because of the punishing value of unreciprocated cooperation. As a result, the number of defectors is also decreasing. Therefore, advanced defectors begin playing based on the frequency to keep cooperators alive. Since advanced defectors receive the cooperation reward, they are able to maintain their energy and increase in size, while naive defectors cannot grow that much because the cooperators around them are dying.

To have a better understanding of the above theory, we have analyzed two additional factors about the agents in the population:

(a) neighbourhood Profile

In the study of neighbourhood profiles for each agent type, we can observe that advanced defectors have more cooperators in their neighbourhood in average when the frequency threshold has a higher value (Figures 5.24, 5.25, and 5.26).



Figure 5.24: The number of cooperators in defectors' neighbourhood in the IG-AD model when the frequency threshold is 0.3, cost of living is 0.5, and S is -2.5. The graph presents the averaged results over 50 simulation runs.



Figure 5.25: The number of cooperators in defectors' neighbourhood in the IG-AD model when the frequency threshold is 0.5, cost of living is 0.5, and S is -2.5. The graph presents the averaged results over 50 simulation runs.



Figure 5.26: The number of cooperators in defectors' neighbourhood in the IG-AD model when the frequency threshold is 0.7, cost of living is 0.5, and S is -2.5. The graph presents the averaged results over 50 simulation runs.

(b) Clusters

Studying clusters in IG-AD model shows that a cluster with more advanced defectors on average is a rising cluster. However, a cluster with more naive defectors on average is a dying cluster or has a slower growth rate (Figures 5.27 and 5.28). This observation confirm our result from **O.5** and its explanation in analysis **A.1**. Figures 5.29a, 5.29b, and 5.29c show an example of a rising cluster in IG-AD model. Figures 5.30a, 5.30b, and 5.30c show an example of a dying cluster in IG-AD model.



Figure 5.27: The number of agents in a rising cluster in IG-AD model when the frequency threshold is 0.3, cost of living is 0.5, and S is -2.5



Figure 5.28: The number of agents in a dying cluster in IG-AD model when the frequency threshold is 0.3, cost of living is 0.5, and S is -2.5



Figure 5.29: An example of a rising cluster in the IG-AD model when the cost of living is 0.5, S is -2.5, and frequency threshold is 0.3. Cooperators are gray and defectors are yellow. Advanced cooperators are blue and advanced defectors are red.



Figure 5.30: An example of a dying cluster in the IG-AD model when the cost of living is 0.5, S is -2.5, and frequency threshold is 0.3. Cooperators are gray and defectors are yellow. Advanced cooperators are blue and advanced defectors are red.

In addition, studying local neighbourhood inside individual clusters show that:

 In a dying cluster, the number of naive defectors in a cooperator's neighbourhood is higher than the number of advanced defectors (Figure 5.31). Therefore, while cooperators are surrounded by naive defectors, there is a higher probability that all the agents in that cluster die.

ii. In a rising cluster, the number of advanced defectors in a cooperator's neighbourhood is higher than the number of naive defectors (Figure 5.32). Therefore, while cooperators are surrounded by advanced defectors, there is a higher probability that all the agents in that cluster survive and cluster grows.



Figure 5.31: An example of cooperators' neighbourhood profile in a dying cluster in the IG-AD model where cost of living is 0.5, S is -2.5, and frequency threshold is 0.3.



Figure 5.32: An example of cooperators' neighbourhood profile in a rising cluster in the IG-AD model where cost of living is 0.5, S is -2.5, and frequency threshold is 0.3.

5.3.4 IG-ACD

In the IG-ACD model advanced defectors play based on the frequency and advanced cooperators play based on the opponent's first game. The number of cooperators in the IG-ACD model is always higher than in the basic model with respect to the different values of S and cost of living (Figures 5.33, 5.35, 5.36, and 5.37). In that case, the number of naive defectors is always higher than the number of advanced defectors (Figures 5.38a, 5.38b, 5.38c, 5.39a, 5.39b, and 5.39c).



Figure 5.33: The number of cooperators (naive and advanced) for different values of S in the IG-ACD model when cost of living is fixed at 0.5 and frequency threshold is 0.3. The graph presents the averaged results over 50 simulation runs.



Figure 5.34: The number of defectors (naive and advanced) for different values of S in the IG-ACD model when cost of living is fixed at 0.5 and frequency threshold is 0.3. The graph presents the averaged results over 50 simulation runs.



Figure 5.35: The number of cooperators (naive and advanced) for different values of cost of living in the IG-ACD model when S is fixed at 0 and frequency threshold is 0.3. The graph presents the averaged results over 50 simulation runs.



Figure 5.36: The number of cooperators (naive and advanced) for different values of S in the IG-ACD model when cost of living is fixed at 0.5 and frequency threshold is 0.7. The graph presents the averaged results over 50 simulation runs.



Figure 5.37: The number of cooperators (naive and advanced) for different values of cost of living in the IG-ACD model when S is fixed at 0 and frequency threshold is 0.7. The graph presents the averaged results over 50 simulation runs.



Figure 5.38: The number of advanced defectors and naive defectors in the IG-ACD models when S is 0, cost of living is 0.5, and frequency threshold takes different values. The graph presents the averaged results over 50 simulation runs.



Figure 5.39: The number of advanced defectors and naive defectors in the IG-ACD models when S is 0, cost of living is 1, and frequency threshold takes different values. The graph presents the averaged results over 50 simulation runs.

Comparing Figures 5.10, 5.33 and 5.36 for different values of S, and Figures 5.11, 5.35, and 5.37 for different values of cost of living, shows that there is no visible

difference between the average number of cooperators in the IG-AC and IG-ACD model. As long as a portion of cooperators are able to tactically base their strategy on the collected information, cooperation evolves with or without the presence of advanced defectors.

In addition, studying the neighbourhood profiles and individual clusters for the IG-ACD model shows the process of the evolution of cooperation in the same way we observed in the IG-AC model.

5.4 Performance Comparisons

Observing the results from different models, we can identify different regions with respect to the outcomes of evolution. Those regions are separated from each other based on the values of cost of living and cost of unreciprocated cooperation, namely:

- 1. A region where the number of cooperators is higher than the number of defectors (C > D).
- 2. A region where the number of cooperators reached the population capacity and all the defectors died (All C = Cap).
- 3. A region where only cooperators survived, but they did not reach the population capacity (All C < Cap).
- A region where the number of cooperators is lower than the number of defectors (D > C).
- 5. A region where the number of defectors reached the population capacity and all the cooperators died (All D = Cap).

Comparing the results from the basic model and IG-ACD model with frequency threshold of 0.3, we can conclude that the IG-ACD model outperforms the basic model. Figures 5.40 and 5.41 present the outcomes of evolution in the described regions. The regions have been captured at round 5000, where the population is stable (after this point, there are just a few changes in the number of agents which does not impact on different regions' formation). Comparing the results (in the represented parameter space) show that:

- 1. There is no 'All D' region on the IG-ACD model.
- 2. The 'D > C' region in the IG-ACD model is smaller than in the basic model.
- 3. The 'C > D' region in the IG-ACD model is larger than in the basic model.
- 4. The 'All C' region in the IG-ACD model is larger than in the basic model.
- 5. In the case of high cost of living, only cooperators survive, and that region is the same in both models. However, in the examined experiments with 10000 rounds, cooperators are not able to fulfill the population maximum capacity.

It should be noted that with the higher value of the frequency threshold, like 0.5 or 0.7, a defector acts more like cooperator and that can impact the total number of cooperators in the population. To eliminate that effect in the process of comparison, we adopt 0.3 as the value for frequency threshold.

	Cost of Living							
s	Basic	0	0.5	1	1.5	2	2.5	
	0	D > C	D > C	C > D	C > D	C > D	All C < Cap	
	-0.5	D > C	D > C	C > D	C > D	C > D	All C < Cap	
	-1	All D = Cap	D > C	C > D	C > D	All C = Cap	All C < Cap	
	-1.5	All D = Cap	D > C	C > D	C > D	All C = Cap	All C < Cap	
	-2	All D = Cap	D > C	C > D	C > D	All C = Cap	All C < Cap	
	-2.5	All D = Cap	D > C	C > D	C > D	All C = Cap	All C < Cap	

Figure 5.40: The outcomes of the evolution in the basic model. The results has been averaged over 50 simulation runs at round 5000.

	Cost of Living							
s	IG-ACD0.3	0	0.5	1	1.5	2	2.5	
	0	C > D	C > D	C > D	C > D	C > D	All C < Cap	
	-0.5	C > D	C > D	C > D	C > D	All C = Cap	All C < Cap	
	-1	C > D	C > D	C > D	C > D	All C = Cap	All C < Cap	
	-1.5	C > D	C > D	C > D	C > D	All C = Cap	All C < Cap	
	-2	D > C	C > D	C > D	C > D	All C = Cap	All C < Cap	
	-2.5	D > C	C > D	C > D	All C = Cap	All C = Cap	All C < Cap	

Figure 5.41: The outcomes of the evolution in the IG-ACD model when the frequency threshold is 0.3. The results has been averaged over 50 simulation runs at round 5000.

Considering the main research question of how enabling agents to dynamically adjust their strategies based on proactively collected knowledge of past behaviour of other agents can impact on the evolution of cooperation, we can conclude that:

 cooperation can evolve under the lower threshold of environmental harshness represented by the cost of living in the IG-ACD model compared to the basic model.

- 2. cooperation can evolve under the lower threshold of harshness represented by the cost of reciprocated cooperation in the IG-ACD model compared to the basic model.
- 3. under the higher threshold of harshness represented by the cost of reciprocated cooperation, cooperators prevail over the defectors in the IG-ACD model compared to the basic model.
- under the higher threshold of environmental adversity representing by cost of living, when cost of living is 2.5, cooperation develops in the same way in both models.

Figures 5.42 and 5.43 present the outcomes of evolution in the described regions for the IG-AD model with frequency threshold of 0.3, and the IG-AC model respectively. We can see that only in one region, the IG-AC model outperforms the IG-ACD model. As mentioned before, the presence of advanced cooperators in the population, improves the conditions for the development of cooperation, compared to the basic model, regardless of the presence of advanced defectors. In addition, one can see that there is no visible difference between the basic model and the IG-AD model in some regions. However, as mentioned before, having only advanced defectors in the population increases the total number of defectors, when environment is moderately harsh, compared to the basic model. Also, the number of cooperators is influenced by the presence of advanced defectors in the IG-AD model, compared to the basic model.

	Cost of Living							
	IG-AD0.3	0	0.5	1	1.5	2	2.5	
	0	D > C	D > C	C > D	C > D	C > D	All C < Cap	
	-0.5	D > C	D > C	C > D	C > D	C > D	All C < Cap	
s	-1	All D = Cap	D > C	C > D	C > D	C > D	All C < Cap	
	-1.5	All D = Cap	D > C	C > D	C > D	C > D	All C < Cap	
	-2	All D = Cap	D > C	C > D	C > D	C > D	All C < Cap	
	-2.5	All D = Cap	D > C	C > D	C > D	C > D	All C < Cap	

Figure 5.42: The outcomes of the evolution in the IG-AD model when the frequency threshold is 0.3. The results has been averaged over 50 simulation runs at round 5000.

	Cost of Living							
	IG-AC	0	0.5	1	1.5	2	2.5	
	0	C > D	C > D	C > D	C > D	All C = Cap	All C < Cap	
	-0.5	C > D	C > D	C > D	C > D	All C = Cap	All C < Cap	
s	-1	C > D	C > D	C > D	C > D	All C = Cap	All C < Cap	
	-1.5	C > D	C > D	C > D	C > D	All C = Cap	All C < Cap	
	-2	D > C	C > D	C > D	C > D	All C = Cap	All C < Cap	
	-2.5	D > C	C > D	C > D	All C = Cap	All C = Cap	All C < Cap	

Figure 5.43: The outcomes of the evolution in the IG-AC model. The results has been averaged over 50 simulation runs at round 5000.

5.4.1 Results Validation

Student t-test is a statistical test which is widely used to compare the mean of two groups of samples. It is, therefore, to evaluate whether the means of the two sets of data are statistically significantly different from each other [Welch, 1947].

The t-test statistic value is calculated as shown in equation 5.1 where:

- 1. a and b are two independent samples.
- 2. m_a and m_b represent the means of samples a and b, respectively.
- 3. n_a and n_b represent the sample size of samples a and b, respectively.
- 4. S^2 is an estimator of the common variance of the two samples.

$$t = \frac{m_a - m_b}{\sqrt{\frac{S^2}{n_a} + \frac{S^2}{n_b}}}$$
(5.1)

The common variance of two samples is calculated as shown in equation 5.2.

$$S^{2} = \frac{\sum_{i=1}^{n} (x_{i} - m_{a})^{2} + \sum_{j=1}^{n} (x_{j} - m_{b})^{2}}{n_{a} + n_{b} - 2}$$
(5.2)

We also can use the *Welch t-test*, where the variances of the two groups being compared are different. The formula for the Welch t- test is provide in equation 5.3 where:

- 1. a and b are two independent samples.
- 2. m_a and m_b represent the means of samples a and b, respectively.
- 3. n_a and n_b represent the sample size of samples a and b, respectively.
- 4. S_a^2 and S_b^2 are the variance of samples a and b, respectively.

$$t = \frac{m_a - m_b}{\sqrt{\frac{S_a^2}{n_a} + \frac{S_b^2}{n_b}}}$$
(5.3)

Running both t-test and Welch t-test shows that the results from the IG-models with sample size 20 are significantly different from the results obtained from the basic model with 95% accuracy. In particular, the difference is statically significant in the regions, where IG-models are superior to the basic model with respect to the number of agents. It should be noted that we have 50 runs of each experiment settings which certainly confirms that the results from advanced models outperform the results from the basic model.

To have a better understanding of the significant difference between the results of the different models, we can draw the error bars as presented in Figure 5.44. The error bars in that example indicated the standard deviation. Although one must perform a statistical test to draw a conclusion, we can say since standard deviation error bars do not overlap, results are significantly different.



Figure 5.44: An example of comparing results from the basic and IG-ACD0.3 model when cost of living is 0.5 and S is 0 by interpreting the error bars .

Chapter 6

Conclusions and Future Work

This thesis presents a new model for evaluating the impact of proactive information gathering (IG) upon the evolution of cooperation among self-interested agents. The IG-model inherits its basis from the model introduced by Smaldino et al. [2013], referred to as the basic model. The basic model demonstrates the effect of environmental adversity on the evolution of cooperation in a spatially distributed population of agents with energy-based life cycle, playing the Prisoner's Dilemma (PD) game in neighbourhood encounters. The novel features of the IG-model allow agents to collect information about the past behaviour of other agents and adjust their strategy based on the gathered information and own memories of past encounters. The impact of proactive information gathering on the evolution of cooperation has been investigated in different variations of the IG-model through simulation experiments with the same parameter space as used in the basic model.

Building on the basic model, we have analyzed different approaches to evaluate the impact of information gathering on the evolution of cooperation in a population. We retain the naive cooperators and naive defectors from the basic model. The naive agents always play their native strategy, cooperation or defection, when they are engaged in an interaction with other agents. In addition, we introduce advanced cooperators and advanced defectors. The advanced agents are capable of adjusting their strategy based on their personal memory and gathered information about cooperation profiles of other agents. Our model differs substantially from similarly motivated published research; partly in the specifics of information gathering and use, and partly in the properties of the agent population, its environment, and environmental harshness concepts inherited from the basic model.

Advanced agents use distinct advanced strategies motivated by their short-term and long-term agendas. While the design and investigation of advanced strategies remains an open research dimension, we demonstrate the adequacy of the model by introducing specific strategies of advanced cooperators and defectors and studying through simulation experiments how their addition to the basic model affects the evolution of cooperation in moderately harsh conditions, as represented by the cost of living and the payoff for unreciprocated cooperation.

Advanced agents are introduced gradually, through several auxiliary models. The first step is IG-AC, in which advanced cooperators are present while all defectors are naive. The second step is IG-AD, in which advanced defectors are present while all cooperators are naive. Early experimentation with those two model variations provided useful feedback for the adjustment of strategy designs, leading to IG-ACD, in which both advanced cooperators and advance defectors exist alongside naive agents. The experimental results show that in the parameter space defined by the 'harshness parameters'(the cost of living and the payoff for unreciprocated cooperation), the region in which the cooperators eventually prevail over defectors is substantially larger in IG-ACD than in the basic model. Given the particular choice of IG-based advanced strategies, the results show that the presence of advanced agents demonstrates a strong positive impact of information gathering upon the development of cooperation and its eventual dominance. In a broader sense, the study demonstrates that the model introduced in this thesis can serve as a suitable framework for the study of evolution of cooperation under moderately harsh environmental conditions in the presence of IG agents with adjustable strategies.

The agent's strategic reasoning, based on the proactively gathered information, is an open research dimension. For future work, we can investigate the influence of information gathering on the evolution of cooperation where advanced agents are capable of playing more sophisticated strategies. We can further specify more complex inquiries for gathering the required information which helps an advanced agent to more completely characterize the true identity of its opponent.

In this study, we have simplified our modeling in order to investigate the impact of information gathering upon the evolution of cooperation, without the impact of external factors. Returning to our original objective of designing practical intelligent agents capable of proactively developing cooperation in interaction with other agents, one may need to analyze the design requirements for specific real-world environments.

Bibliography

- C. Athena Aktipis. Know when to walk away: contingent movement and the evolution of cooperation. *Journal of Theoretical Biology*, 231(2):249–260, 2004.
- C. Athena Aktipis. Is cooperation viable in mobile organisms? simple walk away rule favors the evolution of cooperation in groups. *Evolution and Human Behavior*, 32: 263–276, 2011.
- Yu Awaya. Community enforcement with observation costs. Journal of Economic Theory, 154:173–186, 2014.
- Robert Axelrod. The complexity of Cooperation: Agent-based Models of Competition and Collaboration. Princeton University Press, first edition, 1997.
- Robert Axelrod and William D. Hamilton. The evolution of cooperation. *Science*, 211:1390–1396, 1981.
- Robert M Axelrod. The evolution of cooperation. Basic books, 1984.
- Benjamin Bachi, Sambuddha Ghosh, and Zvika Neeman. Prisoner's dilemma with talk. Technical report, Discussion Paper, Tel Aviv, 2010.
- Hélène Barcelo and Valerio Capraro. Group size effect on cooperation in one-shot social dilemmas. *Scientific reports*, 5, 2015.

- Eric Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. Proceedings of the National Academy of Sciences, 99(suppl 3):7280–7287, 2002.
- Michael Bratman. Intention, Plans, and Practical Reason. Harvard University Press, 1987.
- Michael E. Bratman, David J. Israel, and Martha E. Pollack. Plans and resourcebounded practical reasoning. *Computational Intelligence*, 4:349–355, 1988.
- Kurt Brauchli, Timothy Killingback, and Michael Doebeli. Evolution of cooperation in spatially structured populations. *Journal of theoretical biology*, 200(4):405–417, 1999.
- Francesco Buccafurri, Domenico Rosaci, Giuseppe ML Sarnè, and Luigi Palopoli. Modeling cooperation in multi-agent communities. *Cognitive Systems Research*, 5 (3):171–190, 2004.
- Andrew M Colman. Game theory and its applications: In the social and biological sciences. Psychology Press, 2013.
- Étienne Danchin, Luc-Alain Giraldeau, Thomas J Valone, and Richard H Wagner. Public information: from nosy neighbors to cultural evolution. *Science*, 305(5683): 487–491, 2004.
- Michael Doebeli and Christoph Hauert. Models of cooperation based on the prisoner's dilemma and the snowdrift game. *Ecology letters*, 8(7):748–766, 2005.
- Jim E Doran, SRJN Franklin, Nicholas R Jennings, and Timothy J Norman. On cooperation in multi-agent systems. The Knowledge Engineering Review, 12(03): 309–314, 1997.
- Edith Elkind and Evangelos Markakis. Game theoretic foundations of multiagent systems. In Gerhard Weiss, editor, *Multiagent Systems*, chapter 17, pages 811–847. MIT Press, 2013.
- Joshua M Epstein et al. Zones of cooperation in demographic prisoner's dilemma. *Complexity*, 4(2):36–48, 1998.
- LM Floría, C Gracia-Lázaro, J Gómez-Gardenes, and Y Moreno. Social network reciprocity as a phase transition in evolutionary cooperation. *Physical Review E*, 79(2):026106, 2009.
- Chris D Frith and Uta Frith. Mechanisms of social cognition. Annual review of psychology, 63:287–313, 2012.
- Herbert Gintis. Game theory evolving: A problem-centered introduction to modeling strategic behavior. Princeton university press, 2000.
- William D Hamilton. The evolution of altruistic behavior. The American Naturalist, 97(896):354–356, 1963.
- Dirk Helbing. Agent-based modeling. In Social self-organization, pages 25–70. Springer, 2012.
- Yuval Heller and Erik Mohlin. Observations on cooperation. Available at SSRN, 2016.
- Dedi Iskandar Inan and Ratna Juita. Analysis and design complex and large data base using mysql workbench. International Journal of Computer Science & Information Technology, 3(5):173, 2011.
- Nicholas R Jennings. Controlling cooperative problem solving in industrial multiagent systems using joint intentions. *Artificial intelligence*, 75(2):195–240, 1995.

- Sarit Kraus. Negotiation and cooperation in multi-agent environments. Artificial intelligence, 94(1-2):79–97, 1997.
- Sean Luke, Claudio Cioffi-Revilla, Liviu Panait, Keith Sullivan, and Gabriel Balan. Mason: A multiagent simulation environment. Simulation, 81(7):517–527, 2005.
- Stephen Majeski, Greg Linden, Corina Linden, and Aaron Spitzer. A spatial iterated prisoners dilemma game simulation with movement. In *Simulating Social Phenom*ena, pages 161–167. Springer, 1997.
- Mojdeh Mohtashemi and Lik Mui. Evolution of indirect reciprocity by social information: the role of trust and reputation in evolution of altruism. *Journal of Theoretical Biology*, 223(4):523–531, 2003.
- AB MySQL®. MySQL5.7 reference manual. https://dev.mysql.com/doc/refman/ 5.7/en/, 2017.
- Martin A. Nowak. Five rules for the evolution of cooperation. *Science*, 314:1560–1563, 2006.
- Martin A Nowak and Karl Sigmund. Evolution of indirect reciprocity by image scoring. Nature, 393(6685):573–577, 1998.
- Martin A Nowak, Corina E Tarnita, and Tibor Antal. Evolutionary dynamics in structured populations. *Philosophical Transactions of the Royal Society of London* B: Biological Sciences, 365(1537):19–30, 2010.

Jonathan Asante Nyantakyi. Msc thesis [in progress]. UNBC, 2018.

Oxford University Press. Oxford dictionaries online. https://en. oxforddictionaries.com/definition/toroidal, 2017.

- Parag C Pendharkar. Game theoretical applications for multi-agent systems. *Expert* Systems with Applications, 39(1):273–279, 2012.
- Stuart J. Russell and Peter Norvig. Artificial Intelligence A Modern Approach. Prentice Hall, 1995.
- Stuart Jonathan Russell and Peter Norvig. Artificial intelligence: a modern approach (3rd edition), 2009.
- Jordi Sabater and Carles Sierra. Reputation and social network analysis in multi-agent systems. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: Part 1*, pages 475–482. ACM, 2002.
- Norman Salazar, Juan A Rodriguez-Aguilar, Josep Ll Arcos, Ana Peleteiro, and Juan C Burguillo-Rial. Emerging cooperation on complex networks. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume* 2, pages 669–676. International Foundation for Autonomous Agents and Multiagent Systems, 2011.
- Mahendra Sekaran and Sandip Sen. To help or not to help. In Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society, pages 736–741, 1995.
- Yoav Shoham. Agent-oriented programming. Artificial intelligence, 60(1):51–92, 1993.
- Yoav Shoham and Kevin Leyton-Brown. *Multiagent systems : algorithmic, gametheoretic, and logical foundations.* Cambridge University Press, 2009.
- Karl Sigmund and Martin A Nowak. Evolutionary game theory. Current Biology, 9 (14):R503–R505, 1999.

- Paul E. Smaldino and Jeffrey C. Schank. Movement patterns, social dynamics, and the evolution of cooperation. *Theoretical Population Biology*, 82:48–58, 2012.
- Paul E. Smaldino, Jeffrey C. Schank, and Richard McElreath. Increased costs of cooperation help cooprators in the long run. *The American Naturalist*, 181:451– 463, 2013.
- John Maynard Smith. *Evolution and the Theory of Games*. Cambridge University Press, 1982.
- John Maynard Smith and G. R. Price. The logic of animal conflict. *Nature*, 146: 15–18, 1973.
- Ian Sommerville. Software Engineering. Pearson, 2011.
- György Szabó and Gabor Fath. Evolutionary games on graphs. Physics reports, 446 (4):97–216, 2007.
- Satoru Takahashi. Community enforcement when players observe partners' past play. Journal of Economic Theory, 145(1), 2010.
- Robert E Tarjan. Depth-first search and linear graph algorithms. SIAM journal on computing, 1(2):146–160, 1972.
- Karl Tuyls and Kagan Tumer. Multiagent learning. In Gerhard Weiss, editor, Multiagent Systems, chapter 10, pages 423–475. MIT Press, 2013.
- John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University press, 1944.
- Jörgen W. Weibull. Evolutionary Game Theory. MIT Press, first edition, 1997.
- Bernard L Welch. The generalization of student's' problem when several different population variances are involved. *Biometrika*, 34:28–35, 1947.

- David Sloan Wilson and Daniel Tumminelli O'Brien. Evolutionary theory and cooperation in everyday life. In Simon A. Levin, editor, Games, groups, and the global good, pages 155–168. Springer, 2009.
- David Sloan Wilson and Edward O. Wilson. Rethinking the theoretical foundation of sociobiology. The Quarterly Review of Biology, 82:327–348, 2007.
- Michael Wooldridge. An Introduction to MultiAgent Systems. Wiley, second edition, 2009.
- Michael Wooldridge and Nicholas R. Jennings. Intelligent agents: Theory and practice. Knowledge Engineering Review, 10(2):115–152, 1995.