The Structural Evolution of Morality Revisited

Stefan Seil

Munich 2021

The Structural Evolution of Morality Revisited

Stefan Seil

Master's Thesis
at the Faculty of Mathematics,
Informatics and Statistics
at LMU Munich

submitted by
Stefan Seil

Munich, February 26th 2021

Contents

Abstract **ix**

1 Introduction **1**

2 Related Work **3**

3 Theory **5**
3.1 Evolutionary Game Theory . . . 5
3.2 Agent-Based Simulations . . . 6
3.3 The Structural Evolution of Morality . . . 7

4 Model **11**
4.1 Purpose . . . 11
4.2 Structure . . . 12
4.2.1 Agents . . . 12
4.2.2 Games . . . 12
4.2.3 Networks . . . 18
4.2.4 Learning Rules . . . 24
4.3 Process Overview and Scheduling . . . 25
4.4 Design Concepts . . . 27
4.5 Initialization . . . 29
4.6 Input . . . 30
4.7 Output . . . 30
4.8 Submodels . . . 31
4.8.1 Interactions . . . 31
4.8.2 Learning . . . 31
4.8.3 Mutation . . . 31
4.8.4 Stability . . . 32
4.9 Parameter Overview . . . 33

5 Implementation **35**
5.1 Agent-Based Simulation . . . 35
5.2 Random Network Generation . . . 36

5.2.1 Small-World Networks . . . 36
5.2.2 Bounded-Degree Networks . . . 38
5.2.3 Fully Random Networks . . . 43
5.3 Cycle Detection . . . 45
5.4 Verification . . . 45

6 Validation **47**
6.1 Differences to Alexander's Model . . . 47
6.2 Quantitative Validation . . . 49
6.2.1 Prisoner's Dilemma . . . 49
6.2.2 Stag Hunt . . . 53
6.2.3 Bargaining Subgame . . . 54
6.2.4 Ultimatum Subgame . . . 56

7 Sensitivity Analysis **59**
7.1 Methodology . . . 59
7.2 Experimental Setup . . . 60
7.3 Results . . . 63
7.3.1 General . . . 64
7.3.2 Fully Connected Network . . . 72
7.3.3 Prisoner's Dilemma . . . 76
7.3.4 Stag Hunt . . . 78
7.3.5 Bargaining Subgame . . . 80
7.3.6 Ultimatum Subgame . . . 82

8 Discussion **85**
8.1 Findings . . . 85
8.1.1 Practical Insights . . . 85
8.1.2 Structural Constraints . . . 86
8.1.3 Cooperation Incentive . . . 86
8.1.4 Best-Response Reasoning . . . 87
8.1.5 Small-World Networks . . . 87
8.1.6 Sparse Random Networks . . . 88
8.2 Future Work . . . 89
8.2.1 Model . . . 89
8.2.2 Sensitivity Analysis . . . 90

9 Conclusion **91**

Bibliography **93**

Declaration of Authorship **97**

List of Figures

4.1 Prisoner's Dilemma Normal-Form Representation . . . 13
4.2 Prisoner's Dilemma With Cooperation Incentive . . . 14
4.3 Prisoner's Dilemma With Defection Incentive . . . 14
4.4 Stag Hunt Normal-Form Representation . . . 15
4.5 Stag Hunt With Risk Dominance . . . 15
4.6 Bargaining Subgame Normal-Form Representation . . . 16
4.7 Ultimatum Subgame Extended-Form Representation . . . 17
4.8 Ultimatum Subgame Strategies . . . 17
4.9 Game-Related Parameters . . . 18
4.10 Interaction and Learning Graphs With Learning Distance . . . 19
4.11 Fully Connected Network . . . 19
4.12 One-Dimensional Lattice . . . 20
4.13 Two-Dimensional Lattice Neighborhood Values . . . 21
4.14 Two-Dimensional Lattice . . . 21
4.15 Small-World Network . . . 22
4.16 Bounded-Degree Network . . . 22
4.17 Fully Random Network . . . 23
4.18 Network-Related Parameters . . . 23
4.19 Game Strategy Orders . . . 24
4.20 Model Schedule . . . 26
4.21 Probability Density Functions for Initialization . . . 29
4.22 Configuration Parameters . . . 30
4.23 Neighborhoods With Mutation Distance . . . 32
4.24 All Model Parameters . . . 34

5.1 Replacement Strategies for Replaced Configuration Model . . . 39
5.2 Loop Triple Replacement for Replaced Configuration Model . . . 42
5.3 Comparison of Bounded-Degree Network Generation Algorithms . . . 43

6.1 Prisoner's Dilemma Validation Results for 15 Agents . . . 50
6.2 Prisoner's Dilemma Validation Results for 30 Agents . . . 51
6.3 Prisoner's Dilemma Validation Results for 60 Agents . . . 52
6.4 Stag Hunt Validation Results . . . 53

6.5 Bargaining Subgame Validation Results 55
6.6 Ultimatum Subgame Validation Results for Deactivated Mutation 57
6.7 Ultimatum Subgame Validation Results for Activated Mutation 58

7.1 Parameter Values for Sensitivity Analysis 61
7.2 Small-World Beta Values for Corresponding ω Measures 63
7.3 General: Initial Moral Mean 65
7.4 General: Mutation Distance and Mutation Probability 66
7.5 General: Learning Distance 67
7.6 General: Learning Rule 68
7.7 General: Neighborhood 69
7.8 General: Beta 70
7.9 General: Degree Interval 71
7.10 Prisoner's Dilemma: Fully Connected Network 72
7.11 Prisoner's Dilemma: Replicator Dynamics 72
7.12 Stag Hunt: Fully Connected Network 73
7.13 Bargaining Subgame: Fully Connected Network and Replicator Dynamics . . . 74
7.14 Ultimatum Subgame: Fully Connected Network 75
7.15 Prisoner's Dilemma: Baseline Results 76
7.16 Prisoner's Dilemma: Cooperation Incentive and Defection Incentive . . . 77
7.17 Prisoner's Dilemma: Cooperation Incentive on One-Dimensional Lattice . . . 77
7.18 Stag Hunt: Baseline Results 78
7.19 Stag Hunt: Learning Rule 79
7.20 Stag Hunt: Risk Dominance and Beta 79
7.21 Bargaining Subgame: Baseline Results and Learning Rule 80
7.22 Bargaining Subgame: Population Size on One-Dimensional Small-World . . . 81
7.23 Bargaining Subgame: Beta 81
7.24 Ultimatum Subgame: Baseline Results and Mutation Probability 82
7.25 Ultimatum Subgame: Population Size and Neighborhood 83
7.26 Ultimatum Subgame: Beta 83

Abstract

In his book *The Structural Evolution of Morality* (Alexander 2007), J. McKenzie Alexander uses agent-based simulations to investigate the emergence of social norms in populations of boundedly rational individuals playing games of social interaction on different network topologies. In order to get more insight into which conditions are necessary for moral behaviour to emerge, this paper offers a new agent-based model to build upon Alexander's approach. Numerical methods are used to analyze the influence of different parameters on the emergence of moral behaviour. By focusing on supporting arbitrary combinations of games, network topologies and other parameters, the influence of the individual components on the results can be uncovered using a sensitivity analysis. The *one-factor-at-a-time* method is used to provide a number of interesting results. While the shape of the interpersonal decision problem is still most important, different network topologies can increase the likelihood that moral behaviour emerges in the populations. Cooperation in the Prisoner's Dilemma can only be achieved consistently by strongly incentivizing cooperation in the payoff matrix. The *Best Response* learning rule, which models higher cognitive abilities on part of the individuals, does not only have a negative impact on the emergence of morality in the Stag Hunt, but also in the Bargaining Subgame. Small-world networks are detrimental to the evolution of fairness in the Bargaining Subgame, and do not have a strong influence in the other cases. Sparse random network topologies exhibit a unique behaviour that makes it more likely for a norm of retribution to foster in populations playing the Ultimatum Subgame. Both the model and the results of the analysis can be built upon to further uncover the precise conditions needed for moral behaviour to evolve.

Chapter 1

Introduction

The Structural Evolution of Morality is a theory about the evolutionary emergence of moral behaviour by J. McKenzie Alexander (Alexander 2007). According to this theory, moral behaviour evolves due to boundedly rational individuals engaging in social interactions that can be modelled using game theory. These interactions take place in social networks which put constraints on who can interact with whom. The individuals try to maximize their expected utility in the interactions, and adapt their strategies accordingly over time. This goal of utility maximization creates an evolutionary selection pressure which favours strategies that are more successful at achieving a good payoff in the interactions. Given the right constraints, it can lead populations to a state in which the strategies which we commonly interpret to be morally right—like playing *Cooperate* in the Prisoner's Dilemma or fairly dividing a resource in the Bargaining Game—are precisely the strategies that maximize the expected utility of the individuals. The evolutionary dynamics thus select for moral behaviour among populations of boundedly rational individuals. The *structural* part of the theory is the hypothesis that different social network topologies have different impacts on whether this emergence of morality takes place. In his book, Alexander analyzes this theory with agent-based simulations. He provides many analytically derived results for selected configurations of his model, and shows that moral behaviour can indeed emerge as a stable equilibrium in some cases. Considering that this doesn't happen in every case, though, and that the emergence of such equilibria is indeed quite rare for many of the analyzed interactions, the crucial question is what exactly the conditions look like in which moral behaviour can evolve.

This paper builds upon Alexander's results by taking a more quantitative approach, which can get additional insights about these conditions. This is done by constructing a model that extends Alexander's original model with a focus on systematically combining different parameters and the ability to analyze arbitrary combinations of parameters with common methods from the agent-based modelling literature. Using this new model, a *sensitivity analysis* is carried out, which shows how different parameters of the model (i.e. the conditions) influence the result (i.e. the emergence of moral behaviour). The outcomes of revisiting Alexander's theory in this manner are promising. First, while the emergence of morality is mostly determined by the shape of the decision problem, different network topologies can change these results to different degrees across many parametrizations. Second, incentivizing cooperation is the only

reliable way to foster cooperation in the Prisoner's Dilemma. Third, more sophisticated learning rules (requiring higher cognitive abilities) are very detrimental to morality in both the Stag Hunt and the Bargaining Subgame. Fourth, small-world networks either don't have a strong impact on the results at all, or they provide a negative influence on the emergence of morality. Finally, sparse random network topologies exhibit special behaviour which can have a strong positive impact on the survival of moral strategies in the Ultimatum Subgame. Using these new results from the sensitivity analysis, the investigation of the Structural Evolution of Morality can be taken further by adapting the model accordingly and performing different analyses.

The paper is structured as follows. Chapter 2 gives an overview of related work in the area of agent-based simulations of evolutionary game theory and moral behaviour. Chapter 3 presents some background to the theories this paper builds upon, namely evolutionary game theory, agent-based simulations and Alexander's theory of the Structural Evolution of Morality. In chapter 4, the revised model is described in detail. Based on this description, other researchers should be able to replicate the simulations described in this paper. Chapter 5 then explains some important aspects of the implementation of the model, with a focus on the generation of random social network topologies. Chapter 6 attempts to validate the model against Alexander's model, by replicating some of Alexander's experiments using quantitative data reported in his book. Chapter 7 presents the methodology of the sensitivity analysis and the experimental setup for executing it, and then reports the results of the analysis. In chapter 8, the results of the sensitivity analysis are discussed in the context of Alexander's theory, and some suggestions for future work on this topic are presented. Chapter 9 concludes the paper by summarizing the important points.

Chapter 2

Related Work

Parts of this section are taken from Seil (2020, section 2.3).

J. McKenzie Alexander's *The Structural Evolution of Morality* (Alexander 2007) can be seen to follow a line of research using evolutionary game theory to explain different social and moral phenomena. In more recent times, this scientific project has been brought to the popular science audience through works by Nowak and Highfield (2011) as well as Bowles and Gintis (2013). There has been a sizeable amount of academic work in this area already. In *The Evolution of Cooperation*, Robert Axelrod famously investigated cooperative behaviour in the repeated Prisoner's Dilemma by letting different strategies compete against each other in a computer-based tournament (Axelrod 1984). In his two-volume work *Game Theory and the Social Contract*, Ken Binmore advocates for evolutionary game theory as a systematic tool for investigating ethics, and uses the theory to argue about moral and political philosophy in the tradition of Hume, Rawls and Harsanyi (Binmore 1994; Binmore 1998). Brian Skyrms analyzes moral and social phenomena such as justice and altruism with evolutionary game theory in his work *Evolution of the Social Contract*, where he uses the replicator dynamics to investigate the evolutionary dynamics of these behaviours (Skyrms 1996). Alexander's addition to this lineage is that he uses agent-based models to study the structural properties of a variety of interpersonal decision problems, which can lead to the emergence of some common moral intuitions we have today.

One can find many publications using computational models to analyze the emergence of certain equilibria in interpersonal decision problems constrained by social networks. In fact, they are too numerous and too broadly scattered among different scientific disciplines to be listed conclusively. Following, a few examples for some classic game-theoretic games are given. The Prisoner's Dilemma is chosen disproportionately often, leading to research into the emergence of cooperation on many topologies, including lattices (Ifti, Killingback, and Doebeli 2004), small-world networks (Masuda and Aihara 2003) and dynamic networks which react to the interactions inside the population (Spiekermann 2009). For the Stag Hunt, one can find simulations of the game on dynamic network topologies (Starnini et al. 2011) as well as lattice-based, ring-based and small-world networks (Zhou et al. 2018). The Ultimatum Game was investigated on small-world networks (Xianyu 2010) as well as dynamic networks (Deng, Tang, and

Zhang 2011). In the spirit of Alexander's work, Sun, Zhao, and Robaldo (2017) create an agent-based model to analyze a variation of the Hawk-Dove game on a variety of different network topologies.

There are two points to note about the aforementioned publications. First, they mostly focus on one interpersonal decision problem and/or network topology at a time. Alexander's holistic approach, i.e. using an agent-based model to study a variety of different games of social interaction on different social network topologies, is quite rare. The reasons are apparent: Constructing and analyzing such a model takes much technical skill and time, both of which are oftentimes not available. Second, the publications use many different assumptions, parameters and methods of analysis for their respective models. This makes it hard to compare results, because a common foundation among the different models is lacking. The approach taken in this paper attempts to mitigate this problem. By building a model which can be used with any combination of pre-defined games and network topologies, the results of different parametrizations can be properly compared with one another. Additionally, by keeping the new model close to Alexander's model, the results in this paper can be compared to Alexander's findings with higher certainty.

Chapter 3

Theory

This chapter provides a brief overview of the relevant concepts and theories that this paper builds upon. Section 3.1 delivers a short introduction into evolutionary game theory, which represents the foundation of both Alexander's theory and the model described in this paper. Section 3.2 describes the concept of agent-based simulations, and the benefits they provide for modelling systems governed by evolutionary game theory. Section 3.3 presents the central points of Alexander's theory, *The Structural Evolution of Morality* (Alexander 2007), which delivers the groundwork for this thesis.

3.1 Evolutionary Game Theory

This section is taken from Seil (2020, section 2.2).

Evolutionary game theory provides the foundation of the theory of the Structural Evolution of Morality and the model described in chapter 4. Giving a conclusive introduction into evolutionary game theory would be outside the scope of this paper. For a better coverage of the topic, see Jörgen W. Weibull's classic *Evolutionary Game Theory* (Weibull 1995). This section briefly presents the basic principles of the theory, as some of the points are going to be important for the rest of this paper.

In short, evolutionary game theory can be viewed as the study of infinitely repeated interpersonal decision problems, faced by populations of boundedly rational individuals, whereby some evolutionary selection processes govern the behaviour of those individuals over time. Let us take a classic example from evolutionary biology to make this clearer. Consider W. D. Hamilton's explanation for why populations of mammals have approximately equal ratios of males and females (Hamilton 1967). The individuals of such a population face the interpersonal decision problem of reproduction, whereby each individual embodies the behaviour of having a tendency to produce more males, or that of having a tendency to producing more females. Here, the behaviour of an individual is fixed from birth, but this need not be the case in general. Let us now assume that the payoffs of the decision problem, i.e. reproduction, are the expected numbers of grandchildren an individual is going to have. If there are fewer males than females

among the population, the behaviour of producing more males is then advantageous. A male has higher prospects for mating, which is going to lead him to create more offspring. His parents then get a higher payoff (expected number of grandchildren) in the decision problem of reproduction. The behaviour of producing more males is then going to spread genetically, until there is no relative advantage to producing males anymore. The same holds for females, respectively, when males dominate the population. Thus, over time, the evolutionary processes are going to move the population to a state where the amount of males and females in the population are approximately equal.

There are two main areas of interest to evolutionary game theory (Alexander 2019, sec. 2). For one, we can use the theory to analyze the stability of certain behaviours among a population at a specific point in time. The classical approach to this is the notion of an evolutionarily stable state, which specifies that a population of individuals with a certain distribution of behaviours can resist being taken over by individuals with novel behaviours. Secondly, we can investigate the dynamics of the evolutionary processes, i.e. how the behaviours among the population change over time and whether they converge to some stable equilibrium. There are different models to do so. The classical approach lies in using the replicator dynamics, which analyzes how the frequency of certain behaviours changes among a large population in the limit. Another approach is using agent-based computer simulations, where we can model more sophisticated interactions and constraints compared to the replicator dynamics (see section 3.2). Alexander uses such an agent-based model for investigating his theory of the Structural Evolution of Morality (see section 3.3).

While evolutionary game theory started as an application of game theory to evolutionary biology, it has become an especially valuable tool for economics and the social sciences. The analysis of the evolutionary processes has been carried over to explain market forces and cultural dynamics. Therefore, we can interpret the models used in evolutionary game theory in two ways. One is the interpretation of biological evolution, which was used in the example of the sex ratio among mammalian populations above. The other is the interpretation of cultural evolution, whereby we can view the behaviours as beliefs which are changed by the individuals over time. Here, it is clearer how the bounded rationality of the individuals comes into play, as the individuals then do in fact make choices on their own. This interpretation is the main focus of Alexander's theory and the model described in this paper.

3.2 Agent-Based Simulations

Agent-based simulations are a modelling practice for simulating the behaviour of complex systems, which has received strong interest in may different scientific disciplines. An agent-based simulation can be defined as a computer simulation "made up of agents, objects or entities that behave autonomously" and are "aware of (and interact with) their local environment through simple internal rules for decision-making, movement, and action" (Sanchez and Lucas 2002, p. 116). These simulations are thus well-suited to analyze a particular type of system, namely systems which exhibit behaviour that emerges from the interactions of many autonomous entities (Bandini, Manzoni, and Vizzari 2009). Many areas of study are characterized by such sys-

tems, and the applications of agent-based modelling are correspondingly broad. For example, agent-based simulations can be used to study animal societies, social and economic systems as well as transportation infrastructure (Davidsson et al. 2007). The most important general use case for agent-based models is to predict the behaviour of complex systems which share the characteristics just mentioned. But their use goes beyond prediction. They can be employed to explain and illuminate the inner workings of a system under investigation, provide a framework to guide the collection of large amounts of data from real-world systems, or educate people about systems which are already well-understood (Axelrod 1997; Epstein 2008).

In the context of this paper, agent-based modelling is used to provide a more sophisticated way to analyze the dynamics of systems governed by evolutionary game theory. Concretely, the model described in this paper attempts to explain and predict the behaviour of a hypothetical system which models game-theoretic interactions among boundedly rational individuals on social networks. The crucial advantage of using an agent-based simulation instead of simpler models, like the traditional replicator dynamics, is that this is a viable way to study populations of individuals who do not conform to idealistic assumptions (Axelrod 1997). We do not need to assume that individuals have perfect rationality or an all-encompassing view of the population. Rather, they can be configured to adapt their strategies heuristically based on what happens in their environment. Using agent-based simulations lets us add constraints to the system, such as social network topologies, which restrict the interactions that can take place among the individuals. This makes for a more realistic model of social interactions.

The powerful agent-based approach comes with challenges, though. It can be quite hard to properly analyze the output of such models (Lee et al. 2015). Modellers have to make sure to use suitable statistical methods for the experimental setup and analysis of the results, and to explore and constrain the oftentimes large and complex solution space of the simulations. As the complexity of systems analyzed through agent-based models increase, the use of well-established methods to describe the model (Grimm et al. 2006), to validate and verify it (Balci 1994) and to properly analyze its result data (Broeke, Voorn, and Ligtenberg 2016) becomes more and more important.

3.3 The Structural Evolution of Morality

This section is taken from Seil (2020, section 2.3).

In *The Structural Evolution of Morality* (Alexander 2007), Alexander attempts to ground morality in a foundation of evolutionary game theory. The main thesis of the book can be summarized as follows: Moral behaviour is a result of boundedly rational individuals attempting to maximize their expected utility in games of social interaction, whereby these games take place on social networks which themselves shape the outcomes of the interactions. In order to argue for this point, he analyzes four different game-theoretic games which contain strategy profiles that are supposed to model different moral intuitions. The Prisoner's Dilemma can be seen to reflect *cooperation* when players choose the Pareto-efficient outcome of both playing *Cooperate*. The Stag Hunt models a game of *trust*, because players need to trust each

other to choose *Stag* in order to get the highest payoff. The Nash Bargaining Game (also called divide-the-cake) prompts *fairness*, as players can choose an equal split which leaves everybody with the same payoff and doesn't waste any of the resource. The Ultimatum Game offers the possibility of *retribution*, when a player rejects a seemingly unfair demand, even though it is sub-optimal in the short-term. Alexander thus attempts to show through evolutionary game theory that the strategies reflecting these moral intuitions can be selected by evolutionary pressures as the dominant behaviour among a population. This can then offer an explanation for why we hold these moral norms of cooperation, trust, fairness or retribution. For the most part, his attempt succeeds. The only exception is the Ultimatum Game, where he could not convincingly show how strategies encoding a norm of retribution could reliably succeed in the model (see Alexander 2007, ch. 6).

As mentioned earlier, Alexander uses agent-based simulations to analyze the evolutionary dynamics of the social interactions. These simulations provide a more sophisticated model compared to the replicator dynamics, in that they offer more fine-grained control over the interactions between the individuals, as well as the possibility to add certain constraints (see ibid., sec. 2.2). A typical simulation of Alexander's model can be seen to work as follows: A population of agents is randomly initialized with strategies. The agents populate a social network which determines the kind of interactions that can take place. When interacting with another agent, an individual gets a certain payoff respective to the game that is being played. After a round of interactions, an agent can update his strategy by imitating a neighbor who got a better payoff than himself. This way, the distribution of strategies changes over time. The most important difference to simpler models of evolutionary dynamics is the addition of the structural component, hence the *Structural* Evolution of Morality. Whereas the replicator dynamics model assumes that every agent has an equal probability of interacting with any other agent, the social networks used in Alexander's agent-based model constrain the possible interactions by virtue of their network topology. There are four different kinds of network topologies which are investigated in the book (see ibid., sec. 2.2–2.5). Lattice models constrain agents on a grid, whereby an individual can only interact with the individuals surrounding his cell. Small-world networks model a social network in which the majority of people have few social relations, while a handful of well-connected individuals exhibit additional relations across the network. These additional *bridge edges* effectively shorten the path between many of the other agents. Bounded-degree networks limit the amount of relations each agent can have through a lower and an upper limit. Dynamic networks offer the possibility of changing one's social relations over time, which leads to the social network being influenced by the interactions taking place inside of it. The crucial idea is that these different network topologies have different influences on the evolutionary dynamics. While the Prisoner's Dilemma analyzed through the classic replicator dynamics necessarily leads to a state of complete defection (ibid., pp. 56–59), a one-dimensional lattice-based model can result in Cooperate being selected as the dominant strategy (ibid., pp. 71–73). The evolution of morality is therefore *structural*, in that it is a consequence of the structural constraints on the interactions between individuals.

Let us consider a concrete example in order to make it clear how the Structural Evolution of Morality is supposed to work. Consider a population of people facing interpersonal decision problems corresponding to the Prisoner's Dilemma. The social relationships of the people

are structured in such a way that most of the persons have few acquaintances, while a few individuals have relationships with more people. The social network which describes these relationships can thus be thought of as a small-world network. Each person among the population has a strategy for the Prisoner's Dilemma (*Cooperate* or *Defect*), which she uses for her interactions. The people now regularly engage in Prisoner's Dilemmas with their respective acquaintances. When a person has finished an interaction, she gets a payoff corresponding to the payoff structure of the game. The person then investigates her social relationships and looks for someone who got a better payoff than herself in the previous interactions. If she finds such a person, and this person was using a different strategy than her, then she chooses to imitate this strategy for the upcoming interactions. This way, the strategy of an individual changes. Over time, this strategy updating mechanism can lead to the population converging to a stable state, whereby the changes in strategies have come down to a minimum. The evolutionary dynamics have thus selected an equilibrium, which can contain one dominant strategy that the majority of people are using. Depending on a number of factors, the dominant strategy of the population can turn out to be *Cooperate* or *Defect*. Together with psychological mechanisms that are not directly part of Alexander's theory, this stable strategy can then, over time, turn into a moral intuition stating that cooperation is good (when *Cooperate* is dominant) or bad (when *Defect* is dominant).

Chapter 4

Model

This chapter describes the newly constructed model according to the *ODD protocol* (Grimm et al. 2006). ODD is a standard protocol for describing agent-based models, which focuses on three main reference points: an overview of the model, its central design concepts and the details of the model's behaviour. Given that this protocol comes from ecology research, a number of sections have been changed in order to better reflect the different domain of this model and to aid readers who are as of yet unfamiliar with the ODD protocol. The section titled *State Variables and Scales* in the original ODD specification has been renamed to *Structure*, because the structure of the model is more complex than could reasonably be described by a listing of variables. The original subsection *Fitness* in the section *Design Concepts* was renamed to *Utility*, considering that a cultural-evolutionary interpretation of the evolutionary dynamics is assumed. There is an additional section titled *Output*, which describes the measurements taken from the model in more detail, as well as a section *Parameter Overview* at the end, which lists the available parameters of the model in a table.

4.1 Purpose

The purpose of the model is to get more insight into the specific conditions which are necessary for moral behaviour to evolve in the context of the Structural Evolution of Morality (Alexander 2007). Alexander has shown analytically which requirements need to hold for certain cases of his model. These analytical results can be seen as the holy grail of model analysis. However, their use is also limited to rather simple models constrained to few parameters. The approach in this paper is to use common statistical methods to numerically approximate the behaviour of the system described by the model. The advantage to this approach is that it is more scalable. One can combine and extend arbitrarily complex components of the model, and analyze the system using the same statistical methods. The drawback is that the numerical nature of this approach can miss potentially important details about e.g. tipping points of parameter influence or edge cases of the system behaviour. Additionally, the analysis can be quite computationally demanding and is constrained by the available computing power. Still, this approach is a worthwhile addition to the analysis of Alexander's Structural Evolution of Morality, es-

pecially as it enables other researchers to build upon both the model and the analysis using common approaches from the modelling literature.

4.2 Structure

This section lays out the structure of the model system. The model describes a system in which a population of *agents* interact among each other according to *games of social interaction*. The agents occupy vertices on a *social network*, the topology of which puts constraints on what interactions can take place. Over the course of a simulation of the model, the agents use certain *learning rules* to adapt their behaviour, in order to maximize their expected utility in the social interactions. Many of the specific components of these four categories can be configured using multiple parameters. In the following, the categories are described in detail.

4.2.1 Agents

An agent describes a boundedly rational individual who is taking part in the games of social interaction and subsequent learning process. Each agent is characterized by three different variables: a unique *identifier*, the currently used *strategy* for the respective game-theoretic game, and the *payoff score* accumulated in the current generation by engaging in interactions with other agents. An agent's identifier is set once during the initialization of the model (see section 4.5) and doesn't change over the course of the simulation. The payoff score is reset every generation and ultimately only serves to make a decision in the learning process (see section 4.8.2). The state of the simulation in a specific generation is described by a mapping of all agents' identifiers to their strategies, after the adaptations of the learning process of this generation took place.

4.2.2 Games

The model contains four different interpersonal decision problems, i.e. game-theoretic games. They are all two-player games. The Prisoner's Dilemma, the Stag Hunt and the Bargaining Subgame are simultaneous games (i.e. both players decide which strategy to use at the same time), whereas the Ultimatum Subgame is a sequential game (i.e. first one player makes a decision, then the other player). The model uses the technique of *subgame approximation*, which simplifies an interpersonal decision problem by only regarding a subset of the possible strategies. The Bargaining Subgame is a subset of the Nash Bargaining Game, and the Ultimatum Subgame is an approximation to the full Ultimatum Game. The purpose of these approximations is to lower the total amount of strategies available, and thus to reduce the variance of the results, while trying to keep the general strategic dynamics of the game in question. For the purposes of the result computation (see section 4.7) and the sensitivity analysis (see chapter 7), each game specifies a set of strategies which are considered to be the morally right choices. These strategies are defined in the following sections. Note that this is merely done to more easily categorize the results of the model. It is debatable whether the respective strategies are actually the right

choices with regards to normative ethics. This, however, is a question for practical philosophy and is not going to be discussed here.

Prisoner's Dilemma

The Prisoner's Dilemma is a simultaneous game. It offers the strategies *Cooperate* and *Defect*. The moral strategy is considered to be *Cooperate*. The game is characterized by the normal-form representation visualized in figure 4.1. The parameters R (reward), S (sucker), T (temptation) and P (punishment) need to fulfill the constraints $T > R > P > S$ and $\frac{T+S}{2} < R$. The variation used in this model contains two additional parameters: Cooperation Incentive and Defection Incentive. These parameters take positive rational numbers as values and control the scale of the game's parameters R and T. Note that the two parameters cannot be used at the same time, i.e. one can only scale R or T at a time, but not both simultaneously.

	Cooperate	*Defect*
Cooperate	R, R	S, T
Defect	T, S	P, P

Figure 4.1: Normal-form representation of the general Prisoner's Dilemma.

Cooperation Incentive scales the parameter R. When using this parameter, the values for the game's parameters are chosen as shown in equation 4.1. The parameter is supposed to incentivize or decentivize cooperation in the Prisoner's Dilemma. Figure 4.2 shows the normal-form representation of the game for two different values for Cooperation Incentive.

$$\begin{aligned} R &= 1.0 + \text{Cooperation Incentive} \\ S &= 0 \\ T &= max(3.0, 2.0 + \text{Cooperation Incentive}) \\ P &= 1.0 \end{aligned} \tag{4.1}$$

	Cooperate	*Defect*
Cooperate	1.2 1.2	3.0 0
Defect	0 3.0	1.0 1.0

	Cooperate	*Defect*
Cooperate	6.0 6.0	7.0 0
Defect	0 7.0	1.0 1.0

Figure 4.2: Normal-form representation of the Prisoner's Dilemma using the Cooperation Incentive parameter set to 0.2 (left) and 5.0 (right).

Defection Incentive scales the parameter T. When using this parameter, the values for the game's parameters are chosen as shown in equation 4.2. This parameter incentivizes or decentivizes Defection. Figure 4.3 shows normal-form representations of the Prisoner's Dilemma using two different values for Defection Incentive.

$$\begin{aligned} R &= 2.0 \\ S &= 0 \\ T &= 2.0 + \textsc{Defection Incentive} \\ P &= 1.0 \end{aligned} \tag{4.2}$$

	Cooperate	*Defect*
Cooperate	2.0 2.0	2.2 0
Defect	0 2.2	1.0 1.0

	Cooperate	*Defect*
Cooperate	2.0 2.0	7.0 0
Defect	0 7.0	1.0 1.0

Figure 4.3: Normal-form representation of the Prisoner's Dilemma using the Defection Incentive parameter set to 0.2 (left) and 5.0 (right).

Note that when using Cooperation Incentive < 1.0 or Defection Incentive ≥ 2.0, the resulting game is strictly speaking not a Prisoner's Dilemma anymore, because the constraint $\frac{T+S}{2} < R$ is violated.

Stag Hunt

The Stag Hunt is a simultaneous game. It offers the strategies *Stag* and *Hare*. The moral strategy is considered to be *Stag*. The game is characterized by the normal-form representation shown in figure 4.4. The parameters A, B, C and D need to fulfill the constraint $A > B \geq C > D$.

	Stag	*Hare*
Stag	A, A	C, B
Hare	B, C	D, D

Figure 4.4: Normal-form representation of the general Stag Hunt.

The variation in this model contains an additional parameter: Risk Dominance. This parameter is a Boolean truth value and controls the relation between the game's parameters A, B and D. Depending on whether the parameter is set to true or false, the concrete values for the game's parameters are chosen as shown in figure 4.5. The parameter thus controls whether the strategy *Stag* is risk dominant or not. A strategy is considered risk dominant if it delivers the greatest payoff under the assumption that the other player randomly chooses one of the available strategies with equal probability. Thus, the strategy *Stag* is risk dominant if the constraint $A > B + D$ is satisfied.

	Stag	*Hare*
Stag	3.0, 3.0	0, 2.0
Hare	2.0, 0	1.0, 1.0

	Stag	*Hare*
Stag	2.0, 2.0	0, 1.0
Hare	1.0, 0	1.0, 1.0

Figure 4.5: Normal-form representation of the Stag Hunt using the Risk Dominance parameter set to true (left) and false (right).

Bargaining Subgame

The Bargaining Subgame is a simultaneous game. It is a limited case of the Nash Bargaining Game for a resource of 10 units (Alexander 2007, pp. 148–155), containing only the strategies *Demand 4*, *Demand 5* and *Demand 6*. The moral strategy is considered to be *Demand 5*. The normal-form representation of the Bargaining Subgame is shown in figure 4.6. Limiting the total amount of strategies reduces the variance of the result, and enables robust simulations with fewer agents, which again improves the runtime of the sensitivity analysis (see chapter 7). The choice of the specific strategies keeps the general strategic dynamics of the Nash Bargaining Game (demanding less than, equal to, or more than the equal division of the resource). Based on experiments, approximating the full Nash Bargaining Game using the aforementioned three strategies doesn't significantly change the dynamics of the model. In fact, when using the full game, the population frequently reaches an intermittent state in which all agents play *Demand 4*, *Demand 5*, or *Demand 6* anyways. This phenomenon is also hinted at in the quantitative model validation described in section 6.2.3.

	Demand 4	*Demand 5*	*Demand 6*
Demand 4	4 4	5 4	6 4
Demand 5	4 5	5 5	0 0
Demand 6	4 6	0 0	0 0

Figure 4.6: Normal-form representation of the Bargaining Subgame.

Ultimatum Subgame

The Ultimatum Subgame is a sequential game. It is a limited case of the Ultimatum Game for a resource of 10 units, and was adopted from Alexander's experiments (ibid., pp. 204–205). It only contains those strategies which accept and/or demand values of 5 and 9. The extended-form representation of the Ultimatum Subgame is shown in figure 4.7. The available strategies are listed in table 4.8, whereby *S5 (Easy Rider)* and *S7 (Fairman)* are considered to be the moral strategies. For the Ultimatum Game, subgame approximation is especially useful, considering that the full Ultimatum Game for a resource of 10 units would contain prohibitively many strategies. Even when disallowing *Demand 10* and *Accept 0*, the number of distinct strategies is 4608 ($9 \cdot 2^9$). Leaving this much room for the strategic dynamics would require a very large population as well as very long simulations in order to reach any equilibrium. Note that because the game is sequential, the interactions process (see section 4.8.1) needs to be executed twice for each pair of neighboring agents (once in each direction), such that both agents get the opportunity to be on the demanding and on the accepting side.

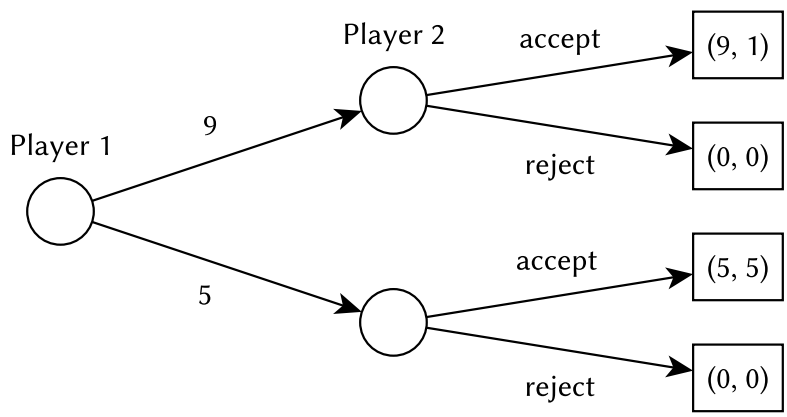

Figure 4.7: Extended-form representation of the Ultimatum Subgame.

Strategy	Demands	Accepts
S1 (Gamesman)	9	5, 9
S2	9	—
S3	9	5
S4 (Mad Dog)	9	9
S5 (Easy Rider)	5	5, 9
S6	5	—
S7 (Fairman)	5	5
S8	5	9

Table 4.8: Strategies of the Ultimatum Subgame.

Overview

The available parameters for the games of the model are listed in table 4.9.

Game	Parameter	Scale
Prisoner's Dilemma	Cooperation Incentive	$(0, \infty)$
	Defection Incentive	$(0, \infty)$
Stag Hunt	Risk Dominance	$\{\text{true}, \text{false}\}$

Table 4.9: Overview of game-related parameters.

4.2.3 Networks

In the context of the model, a network is considered to be the combination of an interaction and a learning graph. Both graphs are undirected and connected. The distinction between the two graphs is done to make it possible for an agent to have differently sized neighborhoods for the interaction and learning processes (see sections 4.8.1 and 4.8.2). Both graphs share the same vertices, but the edges connecting the vertices differ. Each vertex in the two graphs is occupied by one agent. The model contains different network topologies, which impose different structural constraints on the strategic dynamics of the model. The different topologies are mostly adapted from Alexander's experiments (Alexander 2007, pp. 42–48), but have been modified and extended to enable some additional requirements (see section 6.1 for a comparison to Alexander's model). There is an inherit progression among the network topologies from completely regular (lattices), to regular with few randomized connections (small-world networks), to random networks with constraints (bounded-degree networks), to completely random networks (fully random networks). Additionally, there is the fully connected network, which acts as a control topology by imposing no structural constraints at all. The progression of topologies makes it possible to analyze the influence of the regularity or randomness of the social network on the emergence of a stable equilibrium consisting of moral strategies.

All the network topologies share one parameter: Population Size. This parameter simply determines the number of vertices on the graphs, and thus the number of agents occupying the social network. All but the fully connected network share an additional parameter: Learning Distance. This parameter specifies the size of the agents' neighborhoods on the learning graph of the network, based on the topology of the interaction graph. Concretely, it defines the number of depth levels traversed in a breadth-first search from an agent's vertex in the interaction graph of the network. Figure 4.10 visualizes an example for a a simple interaction graph and its corresponding learning graph when setting Learning Distance to 2. Using values larger than one for the Learning Distance expands the agent's survey of neighbors for the adaptive learning process, while keeping the interactions to a smaller locus.

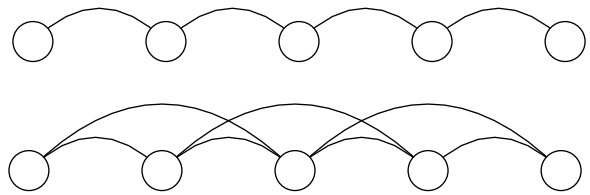

Figure 4.10: Exemplary interaction graph (top) with its corresponding learning graph (bottom) using a Learning Distance of 2.

Fully Connected Network

The fully connected network models a complete graph (i.e. a graph where every vertex is connected with an edge to every other vertex) for both the interaction graph and the learning graph. Figure 4.11 shows an example for a fully connected network. This network type has two purposes. First, it serves as a control for the influence of different network topologies on the strategic dynamics of the model. Second, it is an attempt to imitate the behaviour of the replicator dynamics in an agent-based simulation using social networks. The crucial difference between the two models is that the replicator dynamics assume an infinitely large population, while the fully connected network contains a limited number of agents.

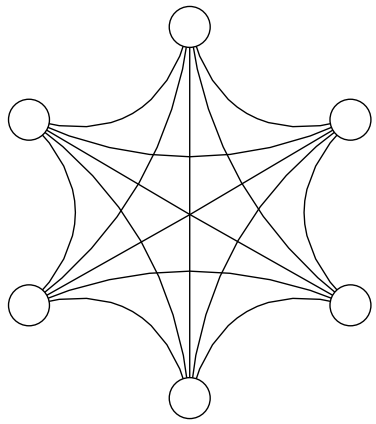

Figure 4.11: Fully connected network with 6 vertices.

Lattice

Lattices are n-dimensional regular network structures. The model supports one- and two-dimensional lattices. Both kinds of lattices have two additional parameters: Neighborhood and Wrap Around.

One-dimensional lattices are vertices connected to form a line. For this kind of lattice, the parameter Neighborhood defines the size of the agents' neighborhoods as the number of steps traversed from the agent in a breadth-first search (exactly like with the Learning Distance parameter). Wrap Around is a Boolean parameter which dictates whether the ends of the lattice are connected by an edge to form a ring. Figure 4.12 shows the interaction graph for an exemplary one-dimensional lattice.

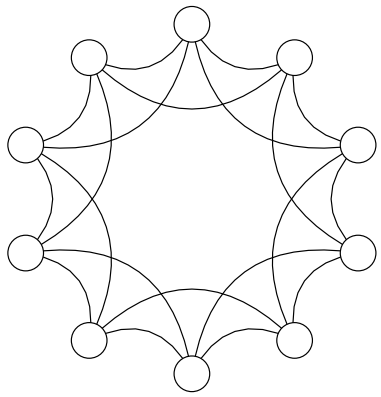

Figure 4.12: One-dimensional lattice with 10 vertices, using Neighborhood = 2 and Wrap Around = true.

Two-dimensional lattices are vertices arranged on a grid. Here, the parameter Neighborhood defines both the kind of neighborhood, as well as its size. There are two different kinds of neighborhoods: the *von Neumann* neighborhood and the *Moore* neighborhood. The von Neumann neighborhood reaches in the four cardinal directions, while the Moore neighborhood additionally includes the four diagonal directions. The kind of neighborhood is coupled with its size, again defined by the depth levels of a breadth-first search starting at an agent's vertex (like with the Neighborhood parameter for the one-dimensional lattice). The possible values for the Neighborhood parameter are listed in table 4.13. The Wrap Around parameter defines whether the edges and corners of the grid wrap around to form a torus. Figure 4.14 shows the interaction graph for an exemplary two-dimensional lattice.

Neighborhood	Type	Size	Agent Count
N4	von Neumann	1	4
N12	von Neumann	2	12
N24	von Neumann	3	24
⋮	⋮	⋮	⋮
M8	Moore	1	8
M24	Moore	2	24
M48	Moore	3	48
⋮	⋮	⋮	⋮

Table 4.13: Possible values for the Neighborhood parameter for two-dimensional lattices.

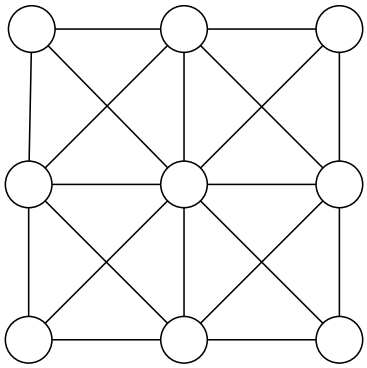

Figure 4.14: Two-dimensional lattice with 9 vertices, using Neighborhood = M8 and Wrap Around = false.

Small-World Network

The small-world networks used in the model are modifications of the lattices described earlier. The modification lies in the random rewiring of a small number of edges on the lattice. This way, the distances between normally far apart regions of the lattice are shortened. This approach conforms to the Watts-Strogatz model for small-world graphs (Watts and Strogatz 1998), with the caveat that the networks used in the model are not limited to one-dimensional ring lattices. Instead, they can take any one- or two-dimensional lattice with arbitrary parametrizations as their basis, forming a one- or two-dimensional small-world network, respectively. This makes it possible to analyze the progression from an arbitrary lattice to its small-world counterpart. Because the small-world networks are based on lattices, they share the Neighborhood and Wrap Around parameters of their one- and two-dimensional lattice counterparts. They have one additional parameter: Beta. This parameter defines the probability with which an edge of the base lattice is randomly rewired. The higher the value chosen for Beta is, the more the resulting network is going to look like a random graph. Figure 4.15 shows exemplary small-world counterparts for the one- and two-dimensional lattices shown earlier.

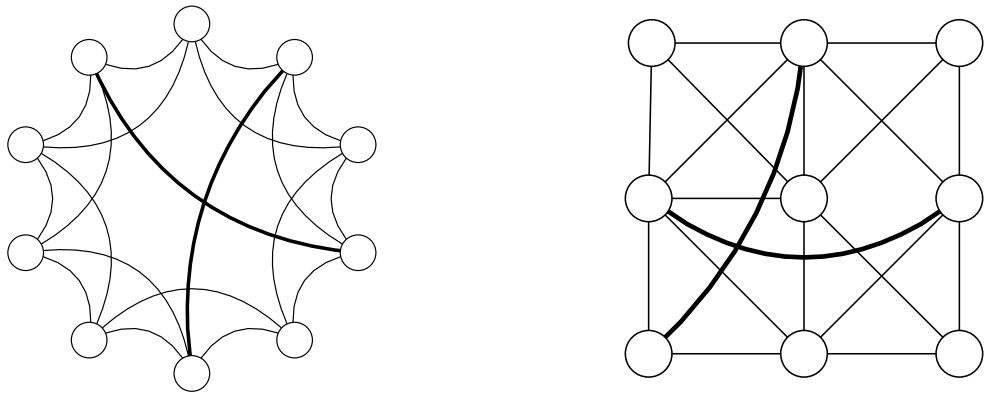

Figure 4.15: Exemplary small-world counterparts for a one-dimensional (left) and two-dimensional (right) lattice. The rewired edges are marked in bold.

Bounded-Degree Network

Bounded-degree networks model graphs in which the degree of each vertex (i.e. the number of edges the vertex is connected to) is limited according to a specified interval $[d_{min}, d_{max}]$. Compared to the fully random network, it preserves some regularity by imposing this constraint on the degrees of the vertices. The network contains one additional parameter: Degree Interval. This parameter specifies the interval $[d_{min}, d_{max}]$, inside of which the degrees of all vertices of the underlying interaction graph have to lie. The degrees are distributed among the vertices according to a uniform probability distribution. Figure 4.16 shows the interaction graph of an exemplary bounded-degree network.

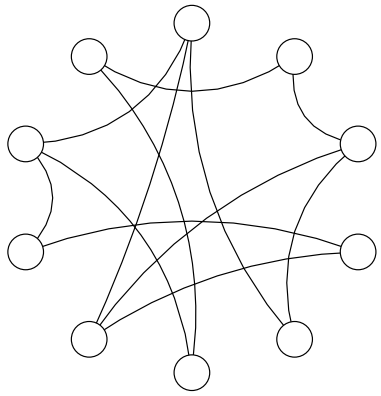

Figure 4.16: Exemplary bounded-degree network with 10 vertices and Degree Interval = $[2, 3]$.

Fully Random Network

The fully random network models a completely random graph. Compared to the bounded-degree network, it doesn't guarantee to preserve any kind of regularity (with the exception that the graph needs to be connected). The network conforms to the Erdős–Rényi model for random graph generation (Erdős and Rényi 1959). It contains one additional parameter: Edge Probability. This parameter defines the probability with which each of the $\binom{N}{2}$ possible edges of a graph with N vertices is added to the interaction graph of the network. Figure 4.17 shows an exemplary fully random network.

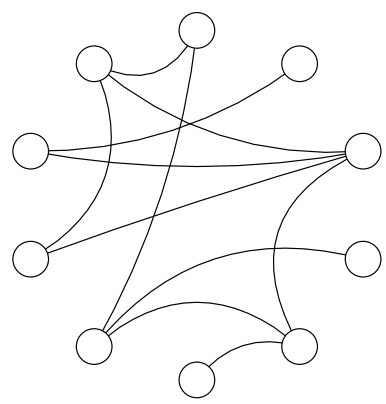

Figure 4.17: Exemplary fully random network with 10 vertices.

Overview

The parameters for the different network topologies of the model are listed in table 4.18.

Network Topology	Parameter	Scale
General	Learning Distance	$[0, \infty)$
	Population Size	$[2, \infty)$
Lattice 1D	Neighborhood	$[1, \infty)$
	Wrap Around	$\{\text{true}, \text{false}\}$
Lattice 2D	Neighborhood	{N4, N12, …, M8, M24, …}
	Wrap Around	$\{\text{true}, \text{false}\}$
Small-World 1D	Beta	$[0, 1]$
Small-World 2D	Beta	$[0, 1]$
Bounded-Degree	Degree Interval	$[[1, \infty), [3, \infty)]$
Fully Random	Edge Probability	$(0, 1]$

Table 4.18: Overview of network-related parameters.

4.2.4 Learning Rules

Learning rules define the mechanism by which agents change their strategies during the learning process of a generation (see section 4.8.2). Each rule specifies whether and how an agent adapts his strategy, based on his neighbors' payoff scores earned during the interaction process. The agent's neighborhood is defined by the network's underlying learning graph (see section 4.2.3). The learning rules which are available in the model are directly taken from Alexander's experiments (Alexander 2007, pp. 39–41). The different rules are supposed to represent a progression from very basic towards more complex decision making, while preserving the bounded rationality assumption about the individuals in the population. The learning rules *Imitate Best*, *Imitate Probability* and *Imitate Average* are quite rudimentary, while *Best Response* requires some more sophisticated cognitive abilities and may stretch the boundaries of the assumptions of evolutionary game theory (ibid., p. 41). The four rules are sometimes abbreviated as *IBest*, *IProb*, *IAvg* and *BestR*, respectively.

When surveying an agent's neighborhood for a better strategy to adopt, it can happen that multiple neighbors using different strategies are equally suited as candidates for imitation. In this case, the strategy of the neighbor with the lowest identifier is chosen among the options. Similarly, when looking at the available strategies directly, there can be multiple strategies which are equally suitable for adoption. In this case, the chosen strategy is the one which comes first in the order of strategies defined by the game being played. The order of the strategies is implicitly defined in the listings of strategies for the different games in section 4.2.2. In table 4.19, the orders are repeated for completion.

Game	Order of Strategies
Prisoner's Dilemma	Cooperate, Defect
Stag Hunt	Stag, Hare
Bargaining Subgame	Demand 4, Demand 5, Demand 6
Ultimatum Subgame	S1 (Gamesman), S2, S3, S4 (Mad Dog), S5 (Easy Rider), S6, S7 (Fairman), S8

Table 4.19: Order of game strategies for conflict resolution.

Imitate Best

Imitate Best represents a very common learning rule in the modelling literature (ibid., p. 39). Using this rule, an agent surveys the payoff scores of all of his neighbors. If no neighbor has a higher score than the agent herself, then she doesn't change her strategy. Otherwise, she imitates the strategy of the neighbor with the highest score in her neighborhood.

Imitate Probability

Imitate Probability adds a stochastic element to the choice of which neighbor to imitate. The higher the payoff of an agent's neighbor is, the more likely she is to imitate this neighbor. Again,

the agent only adapts her strategy if at least one other agent in her neighborhood has a higher payoff score than herself.

Concretely, the learning rule proceeds as follows. Take the set N of all neighbors of the agent a with a higher payoff score than a. Let T be the sum over all score differences between the agent a and the neighbors in N, i.e. $T = \sum_{n \in N}(\text{score}(n) - \text{score}(a))$. Let d_i be the individual score difference between the agent $n_i \in N$ and the agent a. The individual differences d_i in relation to T sum up to a probability distribution, i.e. $\sum_i \frac{d_i}{T} = 1$. The agent a then adopts the strategy of the neighbor n_i with probability $\frac{d_i}{T}$.

Imitate Average

Imitate Average analyzes the average payoff score of each strategy being used in an agent's neighborhood, instead of relying on individual neighbors' scores. The strategy with the highest average payoff score is chosen for imitation.

The learning rule is defined as follows. Take the set of all strategies S available for the game being played. For each strategy $s \in S$, take the subset N_s of N which contains all neighbors that use the strategy s. For each strategy s, calculate the average payoff $p_s = \sum_{i \in |N_s|} \frac{\text{payoff}(n_i)}{|N_s|}$. The agent a then adopts the strategy with the highest average payoff value p_s.

Best Response

Best Response is a slightly more sophisticated learning rule compared to the other ones supported by the model. It adopts the strategy which would have lead to the highest payoff in the current generation. Another way to look at it is that it adopts the strategy which achieves the highest payoff in the next generation, under the assumption that the agent's neighbors don't change their strategies. Thus, it requires at least some counterfactual reasoning on part of the agents (*What would my score have been if I had used this strategy instead?* or *What will my score be in the following generation if my neighbors behave in the same way?*).

4.3 Process Overview and Scheduling

This section provides an overview of the processes which are part of the model. They are described in detail in section 4.8. The model is scheduled in discrete time steps, referred to as generations. During each generation, the four processes *Interactions*, *Learning*, *Mutation* and *Stability* are scheduled in this order. Before the first generation, the initialization process (see section 4.5) is executed to setup the simulation. The simulation is halted by the stability process. The schedule of the model is visualized in figure 4.20.

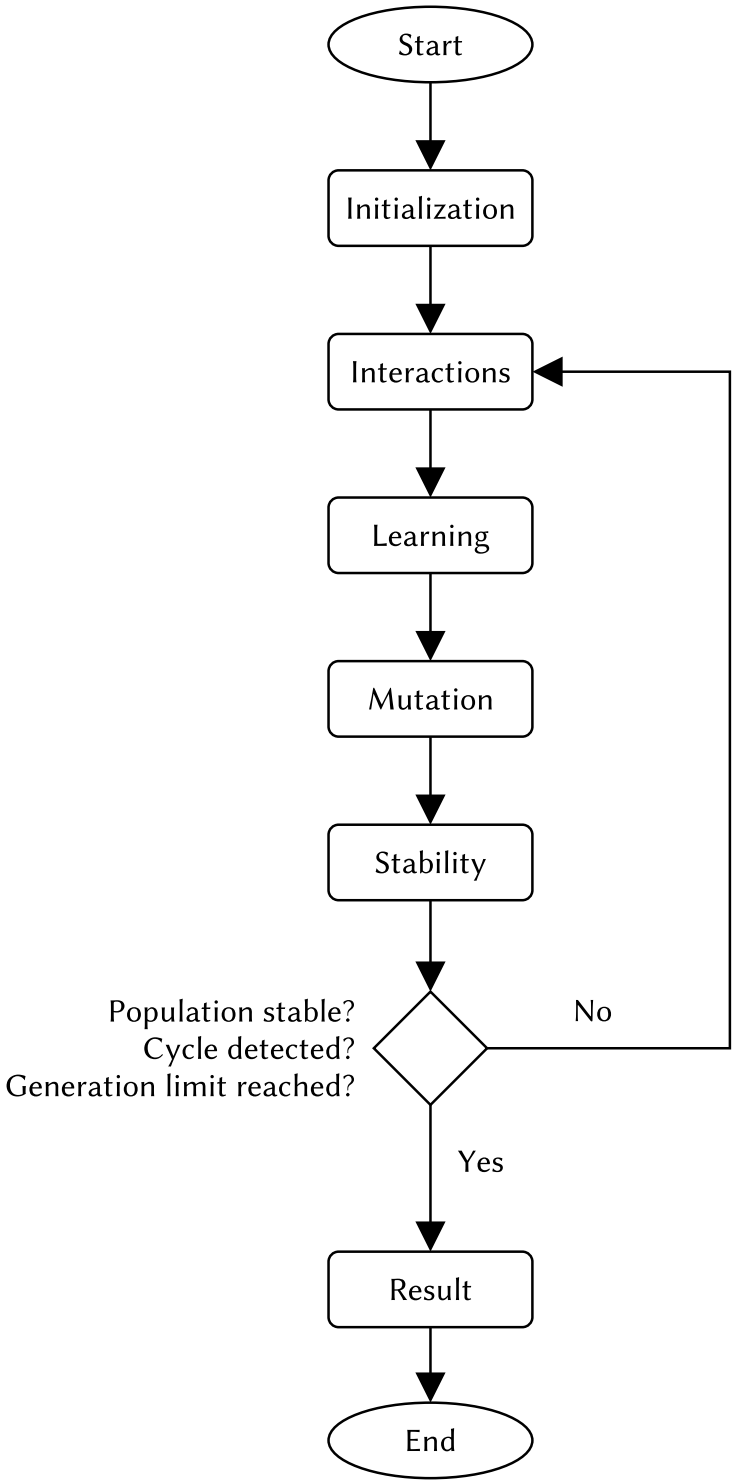

Figure 4.20: Flowchart visualizing the schedule of the model.

Interactions

During this process, the agents engage in the interpersonal decision problem specified by the configured game. The interactions take place between neighbors on the interaction graph of the social network. Each agent collects a cumulative score for the interactions with his neighbors, which is then used in the learning process to adapt the agent's strategy.

Learning

In the learning process, the agents update their strategy using the configured learning rule. To do so, they analyze the payoff scores and strategies of their neighbors in the neighborhood defined by the learning graph of the social network.

Mutation

When the mutation process is activated, each agent gets a change to switch his strategy to a random different one with a specified probability. Agents who mutate can influence their neighbors to adopt the new strategy as well.

Stability

The stability process tracks the state of the population over time. If the population is in a state in which no changes in strategies can take place anymore, or if a cycle of simulation states is detected, this process stops the simulation of the model.

4.4 Design Concepts

This section describes the central design concepts of the model.

Emergence

The strategic dynamics (i.e. the change in the distribution of strategies among the population over time) emerge from the interaction and learning processes over the course of the simulation. The influence of the structural constraints on the strategic dynamics emerges from the specific topology of the social network. Stochastic noise occurs as a results of spontaneous changes of strategies caused by the mutation process. Except for the agents' strategies (and their transient payoff scores), all components are fixed during the initialization and do not change over the course of the simulation.

Adaptation

The learning rules act as adaptive traits, which enable agents to change their strategies in response to their environment. They are chosen to maximize the agents' expected utility in future generations, based on cognitive processes of different sophistication.

Utility

In this model, a cultural interpretation of the evolutionary dynamics is assumed. Thus, the agents attempt to maximize their expected utility using the learning rules as adaptive mechanisms. This is modelled explicitly in the different learning rules used in the model, and the specific way the expected utility is measured depends on the learning rule that is being used (see section 4.2.4). The utility scores which an agent and his neighbors accumulate over the course of the interactions in one generation serve as the basis for the measures of expected utility.

Sensing

The agents are assumed to know their interaction and learning neighborhoods in the social network, the learning rule being used in the model, their own strategy and cumulative utility score for the current generation, as well as the strategies and scores of their neighbors. Which agents are considered neighbors depends on the process an agent takes part in, considering that the learning and interaction neighborhoods can differ from each other. Due to the bounded rationality assumption, there is no prediction of future states or remembering of past states of the simulation. The agents' actions only take the information of the current generation into account.

Interaction

The agents interact according to the game-theoretic interaction defined for the simulation (see section 4.2.2). For the Prisoner's Dilemma, the Stag Hunt and the Bargaining Subgame, the interactions are modelled as one-shot games between an agent and his neighbor. For the sequential Ultimatum Subgame, the agents interact twice with each neighbor.

Stochasticity

There are a number of stochastic elements in the model. The initialization of initial strategies among the population is stochastic, in order to reduce the impact of specific initializations favoring the outcome of the model in one direction or another. The mutation process is stochastic, because it is defined as a random change in strategies among the population. The learning rule *Imitate Probability* uses stochasticity to select which neighbor to imitate, which is supposed to model a boundedly rational approach to imitating a neighbor without systematic or sophisticated reasoning. The small-world networks, bounded-degree networks and fully random networks are generated randomly.

Observation

Throughout the simulation, the generational states of the population (i.e. mappings from agents' identifiers to their respective strategies) are stored in a list. At the end of the simulation, the two result measures *stability* and *morality* are calculated based on this data (see section 4.7).

4.5 Initialization

The initialization of the simulation entails two steps. First, the network is generated based on the parameters for the configured topology (see section 4.2.3). For each vertex in the interaction and learning graph, an agent is created which inhabits this vertex for the rest of the simulation. Second, the initial strategies are distributed among the population. This is done randomly, based on the Initial Moral Mean parameter of the configuration (see section 4.6). At the beginning of the initialization, a probability distribution over the different strategies of the game is calculated. During the distribution of initial strategies, each agent then gets assigned one of the strategies according to the respective probability. Figure 4.21 shows the probability density functions for a moral strategy using two different values for Initial Moral Mean.

The probability distribution is computed like follows. For each strategy s, take the probability density function f_s of a normal distribution $\mathcal{N}(\mu, \sigma^2)$ with $\sigma = 0.1$. If s is considered to be a moral strategy (see section 4.2.2), the Initial Moral Mean parameter specifies the mean μ for the normal distribution. Otherwise, the choice for the mean is $\mu = 0.5$. Then, take a sample x of f_s. If the sample lies outside the interval $[0, 1]$, cap it at the ends of the interval, i.e. $x = max(0, min(1, x))$. Let T be the sum over the samples of all strategies, i.e. $T = \sum_i x_i$. The samples x_i in relation to T sum up to a probability distribution, i.e. $\sum_i \frac{x_i}{T} = 1$. The probability p_s for the strategy s is then defined to be $\frac{x_i}{T}$.

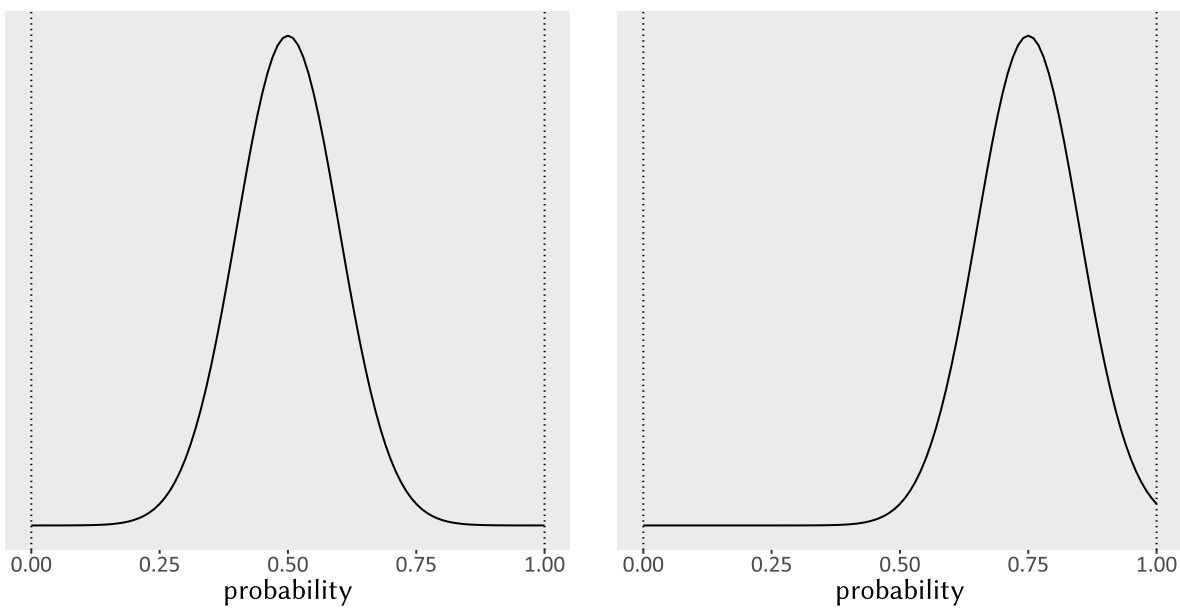

Figure 4.21: Probability density functions for a moral strategy using Initial Moral Mean = 0.5 (left) and Initial Moral Mean = 0.75 (right).

4.6 Input

The input to the model is a configuration which specifies the different components to be used for the simulation, as well as the precise behaviour of the processes of the model. This configuration contains the parameters listed in table 4.22. The parameter Seed specifies the seed for the pseudorandom number generator used for the stochastic elements of the model. When using a non-random social network topology, the results of the model are deterministic with regards to the seed used (see section 5.1 for this limitation).

Parameter	Scale/Values
Game	see section 4.2.2
Network	see section 4.2.3
Learning Rule	see section 4.2.4
Initial Moral Mean	$[0, 1]$
Mutation Probability	$[0, 1]$
Mutation Distance	$[0, \infty)$
Homogeneity Detection	$\{\text{true}, \text{false}\}$
Cycle Detection	$\{\text{true}, \text{false}\}$
Max Generation Count	$[1, \infty)$
Seed	—

Table 4.22: Parameters of the input configuration.

4.7 Output

Once the simulation is finished (due to one of the reasons described in section 4.8.4), the results are calculated. There are two distinct result measures: *stability* and *morality*. Both measures are based on the stable population cycle of the simulation (equation 4.3). The stability measure describes the extent to which the distribution of strategies changes over a certain time period. Concretely, it is defined to be the inverse of the size of the stable population cycle (equation 4.4). The morality measure describes the relative number of agents using moral strategies in the stable population cycle (equation 4.5).

$$cycle = \begin{cases} \text{single generation} & \text{if population converged to a single stable state} \\ \text{generations in cycle} & \text{if population ran into a cycle} \\ \text{all generations} & \text{if simulation reached max. number of generations} \end{cases} \tag{4.3}$$

$$stability = \frac{1}{|cycle|} \tag{4.4}$$

$$
\begin{aligned}
A_{moral} &= \{(agent, strategy, generation) \mid generation \in cycle, strategy \text{ is moral}\} \\
morality &= \frac{|A_{moral}|}{\textsc{Population Size} \cdot |cycle|}
\end{aligned}
\tag{4.5}
$$

4.8 Submodels

In this section, the behaviour of the processes listed in section 4.3 is explained in detail.

4.8.1 Interactions

The interaction process operates on all edges of the network's underlying interaction graph. For each edge, the game is being played between the two agents a_i and a_j occupying the two endpoints of the edge. If the game is sequential (only the Ultimatum Subgame), the game is being played twice, once with agent a_i being the first and agent a_j being the second player, and once vice-versa. If the game is simultaneous, the interaction only takes place once between the two agents.

At the beginning of a generation, each agent starts with a payoff score of zero. After an interaction between two agents is finished, the payoffs of the strategy profile chosen by the strategies of both players are added to the agents' respective payoff scores. This way, the agents accumulate scores across all interactions they are part of in one generation. Note that the payoff scores are not normalized with regards to the number of interactions an agent takes part in. This means that a more connected agent (with more neighbors) can get a higher total score during one generation than a less connected agent, even if the former uses a worse strategy than the latter. The cumulative payoff score of each agent is saved for the succeeding learning process.

4.8.2 Learning

The learning process operates on all vertices in the network's underlying learning graph. For each vertex, the configured learning rule is applied using the agent occupying that vertex as well as that agent's neighborhood according to the learning graph. The new strategy for the agent is stored intermittently. After the process has executed the learning rule for each agent in the population, the intermittently stored strategies are applied to all agents respectively. The agents payoff scores are then reset to zero.

4.8.3 Mutation

The mutation process implements a generalized version of the correlated mutation process using influence neighborhoods described by Vanderschraaf and Alexander (2005, pp. 95–96). It enables agents to randomly change their strategies throughout the simulation, thereby adding noise to the evolutionary dynamics. Upon mutating, agents can also influence their neighbors to adopt the same new strategy.

The behaviour of the mutation process is specified by two parameters: Mutation Probability and Mutation Distance. The process operates on all agents in the population. Each agent individually mutates with the probability defined by Mutation Probability. If an agent is selected for mutation, a random strategy from the configured game's set of strategies is drawn which is different from the agent's current strategy. Thus, when mutation occurs, it is always guaranteed to actually change an agent's strategy. The parameter Mutation Distance defines the size of the neighborhood in the network's interaction graph around the selected agent which is to be affected by the mutation. Like with e.g. the Learning Distance parameter, the distance concretely specifies the depth level of a breadth-first search starting at the selected agent. The affected neighborhood of an agent is visualized in figure 4.23 for different values of Mutation Distance.

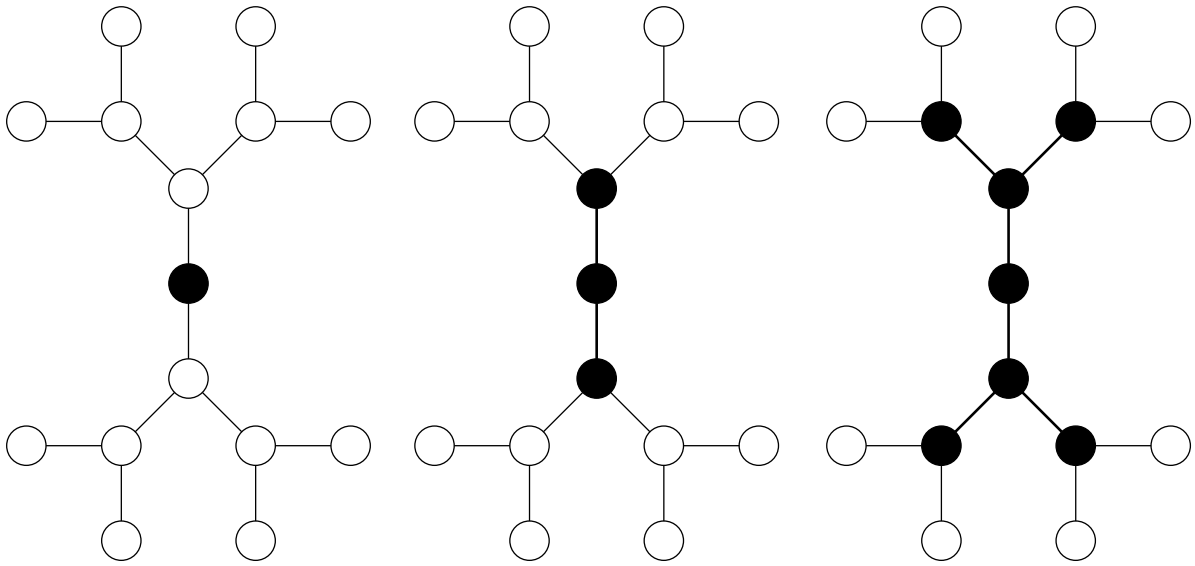

Figure 4.23: Affected neighborhood (in black) of an agent (in the center) when setting Mutation Distance to 0, 1 and 2 (left to right).

4.8.4 Stability

Throughout the simulation, the stability process stores every generation's population state (consisting of a mapping of each agent's identifier to her strategy) in an ordered list. The purpose of the process is to stop the simulation once it has reached a stable state, and to collect the necessary data for the result measure calculation (see section 4.7). Its behaviour is configured by three parameters: Max Generation Count, Homogeneity Detection and Cycle Detection. There are three different scenarios which can be detected:

1. Generation limit is reached (as defined by Max Generation Count)
2. Population state is homogeneous (only if Homogeneity Detection is enabled)
3. Population has run into a cycle (only if Cycle Detection is enabled)

Option 1 is used to limit the runtime of the simulation. The parameter Max Generation Count defines for how many generations the simulation can run before being stopped. Option 2 describes the state in which every agent of the population is using the same strategy. This scenario is only detected if the Boolean parameter Homogeneity Detection is set to true. Option 3 occurs when the changes in strategies among the population are repeating in a fixed pattern indefinitely. It is only detected if the Boolean parameter Cycle Detection is set to true. A cycle is defined as a sequence of succeeding generations limited by an interval $[g_i, g_j)$, whereby the population state of the generation g_i is equal to the population state of the generation g_j. To detect a cycle, the process tries to find the current state of the population in the stored history of population states. Options 2 and 3 are only detected when no stochastic components are configured. Both a non-zero Mutation Probability as well as using the learning rule *Imitate Probability* are considered to specify stochastic components for the model. When mutation is activated by setting Mutation Probability to a value above zero, a stable population state could in principle always change due to random mutations in proceeding generations. When the population has seemingly run into a cycle and *Imitate Probability* is used as the learning rule, agents could change their strategies in ways different than before, moving the population out of the cycle again.

4.9 Parameter Overview

All available parameters of the model are listed in table 4.24.

Parameter	Scale/Values	Process/Component	Description
Game	Prisoner's Dilemma	—	4.2.2
	Stag Hunt	—	
	Bargaining Subgame	—	
	Ultimatum Subgame	—	
Network	Fully Connected	—	4.2.3
	Lattice 1D	—	
	Lattice 2D	—	
	Small-World 1D	—	
	Small-World 2D	—	
	Bounded-Degree	—	
	Fully Random	—	
Learning Rule	Imitate Best	—	4.2.4
	Imitate Probability	—	
	Imitate Average	—	
	Best Response	—	
Cooperation Incentive	$(0, \infty)$	Prisoner's Dilemma	4.2.2
Defection Incentive	$(0, \infty)$	Prisoner's Dilemma	
Risk Dominance	$\{\text{true}, \text{false}\}$	Stag Hunt	
Population Size	$[2, \infty)$	All Networks	4.2.3
Learning Distance	$[0, \infty)$	All Except F. C.	
Neighborhood	$[1, \infty)$	Lattice 1D	
Wrap Around	$\{\text{true}, \text{false}\}$	Lattice 1D	
Neighborhood	$\{N4, N12, \ldots, M8, M24, \ldots\}$	Lattice 2D	
Wrap Around	$\{\text{true}, \text{false}\}$	Lattice 2D	
Neighborhood	$[1, \infty)$	Small-World 1D	
Wrap Around	$\{\text{true}, \text{false}\}$	Small-World 1D	
Beta	$[0, 1]$	Small-World 1D	
Neighborhood	$\{N4, N12, \ldots, M8, M24, \ldots\}$	Small-World 2D	
Wrap Around	$\{\text{true}, \text{false}\}$	Small-World 2D	
Beta	$[0, 1]$	Small-World 2D	
Degree Interval	$[[1, \infty), [3, \infty)]$	Bounded-Degree	
Edge Probability	$(0, 1]$	Fully Random	
Initial Moral Mean	$[0, 1]$	Initialization	4.5
Mutation Probability	$[0, 1]$	Mutation	4.8.3
Mutation Distance	$[0, \infty)$	Mutation	
Homogeneity Detection	$\{\text{true}, \text{false}\}$	Stability	4.8.4
Cycle Detection	$\{\text{true}, \text{false}\}$	Stability	
Max Generation Count	$[1, \infty)$	Stability	
Seed	—	—	4.6

Table 4.24: Overview of all available parameters of the model.

Chapter 5

Implementation

The source code for the model implementation can be found at `https://github.com/srseil/Seom/tree/v1.0.0` *as well as at* `https://pms.ifi.lmu.de/publications/diplomarbeiten/Stefan.Seil/software.zip`.

This chapter describes the solutions to a number of interesting challenges when implementing the model described in chapter 4. Section 5.1 describes some general aspects of the implementation of the model. Section 5.2 delves into the generation of random social networks. Section 5.3 explains the approach to detecting cycles in the stability process of the simulation, and 5.4 lays out the approach to verify the correctness of the implementation.

5.1 Agent-Based Simulation

The implementation of the model was done in Java using the MASON framework for agent-based simulations (Luke, Cioffi, et al. 2005, `https://cs.gmu.edu/~eclab/projects/mason/`) in version 20. For the social networks, the JUNG library (`https://jrtom.github.io/jung/`) was chosen in version 2.1.1 for its high stability and interoperability with the MASON framework. Due to limitations of Java's internal pseudorandom number generator (PRNG), the model uses an implementation of the Mersenne Twister algorithm (`http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html`) provided by the MASON framework in version MT199937(99/10/29) wherever possible.

One goal during the implementation was to make the simulations deterministic for the chosen PRNG seed, such that one could execute the simulation multiple times for a set of parameters and a PRNG seed and get the same results every time. Unfortunately, this goal could not be fully achieved due to the implementation of graphs in the JUNG library. When querying a set of vertices or edges from a graph, the interface of JUNG does not guarantee a stable ordering of the provided results. This can lead to differences in the behaviour of the random network generation. Attempts were made to circumvent this problem by sorting the queried results before continuing the generation, but not all of the indeterminism could be eliminated from the execution. Therefore, the simulations are only deterministic for those configurations which do

not involve a random network (i.e. only for fully connected networks and lattices). All other stochastic processes, however, are fully deterministic for a given PRNG seed.

5.2 Random Network Generation

All networks in the simulation are modelled as undirected multigraphs (i.e. graphs which can contain more than one edge between two vertices) made up of two different kinds of edges: interaction edges and learning edges. This approach makes it straightforward to vary an agent's interaction and learning distance. The implementation can query the interaction and learning subgraph of a network, and use those for the respective interaction and learning processes. The generation of non-random networks (i.e. fully connected networks and lattices) is trivial. The generation of random networks (i.e. small-world networks, bounded-degree networks and fully random networks), however, offers a number of challenges. Most importantly, the generated networks need to be connected graphs, such that the population is not divided into isolated subgraphs which cannot influence each other. Additionally, the generation needs to be performant enough in practice in order not to slow down the initialization process too much.

5.2.1 Small-World Networks

The algorithm for generating small-world networks used in the implementation is a modification of the Watts-Strogatz model for small-world networks (Watts and Strogatz 1998). The original model uses a central parameter β to control the regularity or randomness of the resulting graph. It is presented in algorithm 1.

Algorithm 1 Watts-Strogatz Model

Input: N = vertex count, K = mean degree, β = randomness control
Output: Undirected graph with N vertices and $\frac{NK}{2}$ edges exhibiting small-world properties

Ensure: $0 \leq \beta \leq 1$

```
procedure WattsStrogatz(N, K, β)
    G = one-dimensional ring lattice with N vertices
    for each edge e in G do
        rewired = true with probability β
        if rewired is true then
            disconnect e from its right endpoint v1
            v2 = randomly chosen vertex: v2 ≠ v1
            connect e to v2
        end if
    end for
    return G
end procedure
```

The modification used in the implementation is to accept any one-dimensional or two-dimensional lattice as a basis, including ones which do not wrap around and thus do not form a ring or torus. The crucial advantage of this is that we can compare the progression of every possible lattice supported by the model to its respective small-world counterpart. Considering one can simulate the model with both a specific lattice and its small-world version using the same parameters, the effects of the small-world properties can be filtered out very well. The modification of the Watts-Strogatz model is presented in algorithm 2.

Algorithm 2 Modified Watts-Strogatz Model

Input: L = base lattice, β = randomness control
Output: Undirected graph based on L exhibiting small-world properties

Ensure: $0 \leq \beta \leq 1$

```
procedure WattsStrogatzModified(L, β)
    for each edge e in L do
        rewired = true with probability β
        if rewired is true then
            v1 = randomly chosen vertex from one of the two endpoints of e
            v2 = randomly chosen vertex: v2 ≠ v1 and v2 not connected to v1
            if v2 exists then
                disconnect e from v1
                connect e to v2
            else
                restart WattsStrogatzModified
            end if
        end if
    end for
    if L is not connected then
        restart WattsStrogatzModified
    end if
    return L
end procedure
```

The vertex to be rewired among the existing endpoints of an edge is chosen randomly, such that even the outermost vertex on a non-ring structure (i.e. a lattice that does not wrap around) can be chosen to be rewired. Note that the probability distribution over the vertices is not quite uniform anymore if the underlying lattice is not a ring, because the outermost two vertices have less edges connected to them and thus are chosen with lower probability than the rest. In case a chosen edge cannot be rewired, because there is no other edge which fulfills the requirements, the algorithm is restarted from scratch. Similarly, if the constructed graph is not connected, the generation is started from scratch. Both of these cases happen extremely rarely, though, such that this brute-force method is good enough for the practical execution of the simulations.

5.2.2 Bounded-Degree Networks

The bounded-degree networks used in the simulation are undirected, connected graphs with a fixed number of vertices, whereby the degrees of vertices are uniformly distributed among an interval $[d_{min}, d_{max}]$. The approach to generating networks of bounded degrees is based on a comparison of different models for this purpose described by Britton, Deijfen, and Martin-Löf (2006). The concrete algorithm used is a variation of the *configuration model* described by Bollobás (1980) and Wormald (1980). The original configuration model offers an algorithm for generating graphs of bounded degrees in which the degrees are distributed according to any specified probability distribution. It is shown in algorithm 3.

Algorithm 3 Configuration Model

Input: N = vertex count, F = probability distribution
Output: Undirected graph with N vertices and degrees distributed according to F

```
procedure CONFIGURATION(N, F)
    G = undirected graph with N vertices and no edges
    for each vertex v in G do
        d = random degree chosen independently from F
        attach d stubs to v
    end for
    while stubs are remaining in G do
        s1, s2 = two randomly chosen stubs: s1 ≠ s2
        join s1 and s2 to form an edge
    end while
    return G
end procedure
```

The problem with this algorithm is that it can generate graphs with loop edges (i.e. edges connecting one and the same vertex) and duplicate edges (i.e. more than one edge between two distinct vertices). Britton, Deijfen, and Martin-Löf (2006) describes two approaches to combat this problem: the *erased configuration model* and the *repeated configuration model.* The erased configuration model works by removing any unwanted edges from the generated graph. The repeated configuration model operates by repeating the algorithm until a graph is generated which does not contain any unwanted edges.

Neither of these two solutions is appropriate for the generation of bounded-degree networks for the simulation. Using the erased configuration model, it is not guaranteed that the degree of each vertex lies within the interval $[d_{min}, d_{max}]$. One of the hypotheses of the model is that the connectivity of the social network is quite important for the evolutionary dynamics of the population. Specifically, the difference between e.g. a degree of one and a degree of two could lead to a big difference in the strategic dynamics of the agent occupying that vertex. Therefore, the erased configuration model is not suitable for this use case. The repeated configuration model would in theory be suitable, but turns out to be prohibitively slow for larger values

of $[d_{min}, d_{max}]$ relative to the number of vertices. The connectivity of the graph grows as the degree interval approaches the number of vertices in the graph. The number of unwanted edges produced by the configuration model grows with the connectivity, because more edges offer more potential for loops and for creating an edge between two vertices which are already connected. Hence, the repeated configuration model is not applicable to this use case, either.

To solve the problem, the implementation uses a variation of the configuration model named the *replaced configuration model*. Instead of erasing unwanted edges, this variation attempts to randomly rewire them in the graph such that the formerly chosen degrees do not change. For simplicity, the implementation assumes a uniform probability distribution over the degrees in the interval $[d_{min}, d_{max}]$. The replacement strategies are visualized in figure 5.1, while the whole replaced configuration model is shown in algorithms 4 and 5.

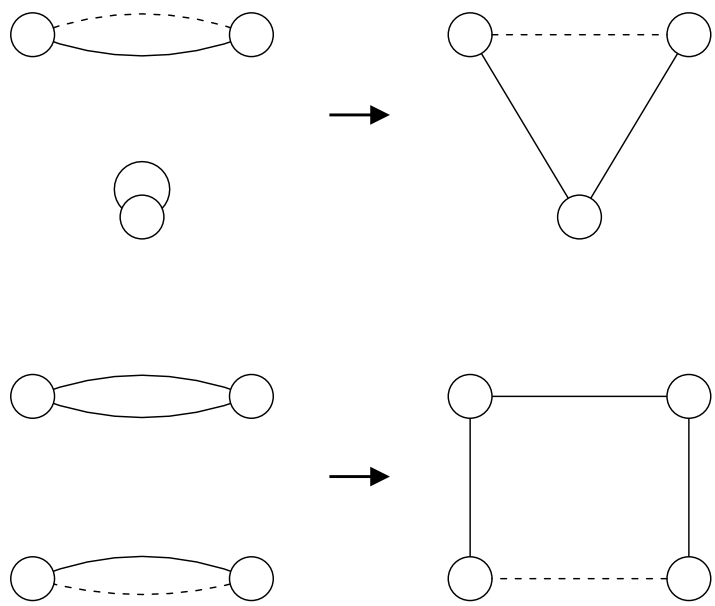

Figure 5.1: Replacement strategies for the replaced configuration model.

Algorithm 4 Replaced Configuration Model (Part 1)

Input: N = vertex count, d_{min} = lower degree bound, d_{max} = upper degree bound

Output: Undirected graph with N vertices and degrees uniformly distributed among $[d_{min}, d_{max}]$

```
procedure ReplacedConfiguration(N, d_min, d_max)
    G = undirected graph with N vertices and no edges

    for each vertex v in G do                                   ▷ Distribute degrees
        d = degree, randomly chosen uniformly from [d_min, d_max]
        attach d stubs to v
    end for

    if number of stubs in G is odd then                         ▷ Ensure the number of stubs is even
        v_1 = randomly chosen vertex with degree > d_min
        if v_r exists then
            remove one random stub from v_1
        else
            v_2 = randomly chosen vertex
            add stub to v_2
        end if
    end if

    while stubs are remaining in G do                           ▷ Connect stubs randomly
        s_1, s_2 = two randomly chosen stubs: s_1 ≠ s_2
        join s_1 and s_2 to form an edge
    end while
```

Algorithm 5 Replaced Configuration Model (Part 2)

```
    E_r = set of edges e in G: e is loop edge or duplicate edge      ▷ Replace unwanted edges
    for each edge e_r in E_r do
        E = set of all edges in G
        rewired = false
        repeat
            e = randomly chosen edge from E
            if e = e_r then
                remove e from E
            else if e is loop edge and e_r is loop edge then
                remove e from E
            else if G contains vertex v: v connected to both e and e_r then
                remove e from E
            else if G contains vertex v: v connected to e_r and
                    v is neighbor of either of the two endpoints of e then
                remove e from E
            else
                v_1, v_2 = endpoints of e
                u_1, u_2 = endpoints of e_r
                remove e and e_r from G
                add edge between v_1 and u_1 to G
                add edge between v_2 and u_2 to G
                rewired = true
            end if
        until rewired is true or E is empty
        if rewired is false then
            restart ReplacedConfiguration
        end if
    end for

    if G is not connected then                      ▷ Restart if graph is not connected
        restart ReplacedConfiguration
    end if

    return G
end procedure
```

During implementation, an additional method for removing unwanted edges was tested: removing loop triples. The method is visualized in figure 5.2, and the corresponding pseudocode is shown in algorithm 6.

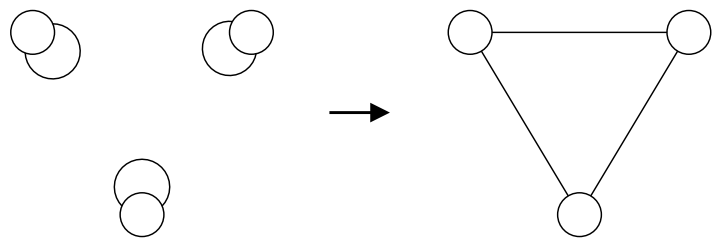

Figure 5.2: Loop triple replacement for the replaced configuration model.

Algorithm 6 Loop Triple Replacement

```
E_l = set of edges e in G: e is loop edge
e_1, e_2, e_3 = randomly chosen edges from E_l
if e_1, e_2, e_3 exist then
    if G contains vertex v: v connected to two edges in {e_1, e_2, e_3} then
        continue
    else if G contains edge e: e connects two endpoints of edges in {e_1, e_2, e_3} then
        continue
    else
        v_1, v_2 = endpoints of e_1
        u_1, u_2 = endpoints of e_2
        w_1, w_2 = endpoints of e_3
        remove e_1, e_2, e_3 from G
        add edge between v_2 and u_1 to G
        add edge between u_2 and w_1 to G
        add edge between w_2 and v_1 to G
    end if
end if
```

Even with the replaced configuration model, there are still plenty of cases in which the generation can fail. If an unwanted edge cannot be removed, or if the resulting graph is not connected, the algorithm has to be started from scratch. However, based on practical experiments, the success rate of the generation is much higher for the replaced configuration model. Table 5.3 shows experimental results for the relative number of failures of the three approaches just discussed. The addition of the loop triple replacement turns out to decrease the performance of the network generation for higher values of d_{min} and d_{max}. Therefore, it was deactivated for

the simulations. One additional problem is that for low values of d_{min} and d_{max} (like in the interval $[1, 2]$), the network generation takes prohibitively long as the number of vertices grows, because the algorithm generates disconnected graphs very frequently. Thus, very low values like these could not be chosen for the simulations of the sensitivity analysis (see chapter 7).

Algorithm	$[2, 3]$	$[3, 5]$	$[5, 8]$
Repeated configuration model	3.534	90.425	∞
Replaced configuration model	0	0.094	28.838
Replaced with loop triple replacement	0	0.094	32.759

Table 5.3: Comparison of bounded-degree network generation algorithms. Shown are the relative numbers of failures for 1000 network generations, using the same PRNG seeds across different algorithms. For the degree interval $[5, 8]$, the generation using the repeated configuration model took prohibitively long to finish and was aborted prematurely.

5.2.3 Fully Random Networks

To generate fully random networks, the implementation uses a modification of what is commonly known as the $G(n, p)$ variant of the Erdős-Rényi model for random graph generation (Erdős and Rényi 1959), even though it was Edgar Gilbert who published this variant independently of Erdős and Rényi (Gilbert 1959). The $G(n, p)$ model creates a randomly wired graph and is shown in algorithm 7.

Algorithm 7 Erdős-Rényi Model

Input: N = vertex count, p = edge probability
Output: Undirected graph with N vertices and $p\binom{N}{2}$ vertices

Ensure: $0 \leq p \leq 1$

```
procedure ErdosRenyi(N, p)
    G = undirected graph with N vertices and no edges
    for each edge e among (N 2) possible edges in G do
        added = true with probability p
        if added is true then
            add e to G
        end if
    end for
    return G
end procedure
```

The modification of the model used in the implementation makes sure to generate a connected graph. The pseudo-code for this modified version is shown in algorithm 8.

Algorithm 8 Modified Erdős-Rényi Model

Input: N = vertex count, p = edge probability
Output: Undirected graph with N vertices and $p\binom{N}{2}$ vertices

Ensure: $0 \leq p \leq 1$

```
procedure ErdosRenyi(N, p)
    G = undirected complete graph with N vertices and (N 2) edges

    E_r = empty set                                  ▷ Attempt to randomly remove edges
    for each edge e in G do
        removed = true with probability 1 − p
        if removed is true then
            v_1, v_2 = endpoints of e
            if v_1, v_2 both have degree ≥ 2 then
                remove e from G
            else
                add e to E_r
            end if
        end if
    end for

    for each edge e_r in E_r do                      ▷ Remove alternatives for leftover edges
        e = randomly chosen edge from G: e not in E_r
        if e exists then
            v_1, v_2 = endpoints of e
            if v_1, v_2 both have degree ≥ 2 then
                remove e from G
                remove e_r from E_r
            end if
        else
            restart ErdosRenyi
        end if
    end for

    if G is not connected then                       ▷ Restart if graph is not connected
        restart ErdosRenyi
    end if

    return G
end procedure
```

5.3 Cycle Detection

When mutation is disabled for a simulation, the population can run into a cycle whereby a fixed series of changes in the strategy distribution repeats indefinitely. In such a case, the simulation should be halted. In order to do this, the implementation includes a mechanism to automatically detect finite cycles based on the previously observed states of the simulation. At the end of each generation t, a hash value h_t of the current state of the simulation is calculated and appended to an ordered list of simulation states. This list is then iterated from the beginning to see if there exists an equal hash value h_i which has been stored in the past generation i. If an equal value is found, the simulation has run into a cycle starting at generation i and ending at generation $t - 1$. Concretely, each available strategy of the game which is being played is assigned a unique integer. The integer representations of the strategies of all agents are then stored in an array, in ascending order of the agents' identifiers. Considering agents cannot be added to or removed from the population, this array serves as a unique representation of the particular state of the simulation. For easier comparison, a hash value of the array is calculated using the cryptographic hash function SHA-256. Java's internal implementation of `hashCode()` is not suitable for this use case, because it produces too many collisions too quickly.

5.4 Verification

To verify the correctness of the implementation, standard software engineering practices were used. The Java programming language provides a static type system which can catch many errors at compile time. The choice of using the well-tested libraries MASON and JUNG made sure to eliminate bugs in both the general execution of the agent-based simulation and in the generation of social networks. Throughout the development of the implementation, the simulations were executed manually in order to see whether the results of the model were plausible. Additionally, the implementation contains unit tests for all of the main components of the model, which verify the correctness of a number of sample cases of the simulations. The unit tests all passed before the final simulations of the sensitivity analysis were carried out. The quantitative validation of the model described in section 6.2 provides further verification of the whole system when using two-dimensional lattices and bounded-degree networks.

Chapter 6

Validation

While there is a wide range of validation techniques applicable to agent-based models (Klügl 2008), many of them are focused on comparing the simulation results to data provided by the real-world system that is being modelled. In the case of the model described in this paper, there is no real-world system that could be readily accessed to provide data for this purpose. Thus, the validation focuses on the approach of *model alignment* (Axtell et al. 1996), which attempts to validate a model by replicating the results of a different model which models the same real-world system. The approach to validation is therefore to replicate the results of Alexander's experiments in *The Structural Evolution of Morality* (Alexander 2007).

There are some important differences between Alexander's approach and the approach chosen in this paper with regards to how the models are constructed and used. For most combinations of games and networks, Alexander focuses on analytically deriving the behaviour of the model using selected parametrizations. Some parameters are only used for specific experiments, e.g. correlated mutation on two-dimensional lattices and bounded-degree networks in the Stag Hunt (ibid., pp. 128–131, 138–142). In contrast, the model described in this paper focuses on using numerical approaches and a systematic combination of parameters. This makes it difficult to validate the model against many of Alexander's results. One would have to extract the respective configurations from the prose text and replicate or approximate them using the parameters provided by the new model. Unfortunately, this is beyond the scope of this paper and has thus been left out. Instead, the validation focuses on those subsets of experiments where Alexander used numerical approaches and reports the associated quantitative data. Thus, this chapter first describes the important differences between the model of this paper and Alexander's model in section 6.1, and subsequently reports the quantitative validation of the new model against Alexander's data in section 6.2.

6.1 Differences to Alexander's Model

There are a number of differences between the model described in chapter 4 and the one used by Alexander. These differences need to be taken into account when attempting to validate the model of this paper against Alexander's model.

As alluded to earlier, one of the central points of the new model is the systematic combination of parameters. Some variations which have already been part of Alexander's experiments have been systematically incorporated as parameters, such that they can be used in a variety of model configurations. The Learning Distance parameter is not limited to lattices, but can be used for all network topologies except the fully connected network (for which it is redundant, because this network models a complete graph). The Neighborhood parameter is applicable not only to lattices, but also small-world networks, and can take on arbitrary sizes. The variations in the payoff matrices of the Prisoner's Dilemma and the Stag Hunt have been consolidated in the parameters Cooperation Incentive, Defection Incentive and Risk Dominance. In Alexander's experiments, the payoff matrices are more arbitrarily chosen and thus offer somewhat more variability, but they still fulfill the general constraints of the Prisoner's Dilemma (Alexander 2007, p. 55). The initialization of the new model is governed by the Initial Moral Mean parameter, while Alexander uses many different approaches to initializing the population state depending on the experiment.

There are some constraints and components which are not part of Alexander's model at all:

- Ensuring all networks are connected graphs
- Extended small-world networks (using arbitrary lattices as bases)
- Fully connected networks (as potential alternative to replicator dynamics)
- Fully random networks
- Cycle detection

Conversely, there are parts of Alexander's model which are not present in the new model:

- Inclusion of replicator dynamics
- Dynamic networks (ibid., pp. 48–52), using the model for social network formation described by Skyrms and Pemantle (2000)
- Decoupled frequencies for learning and interaction processes (Alexander 2007, p. 52), only applicable to dynamic networks
- Past discounting in agents' decision making (ibid., e.g. 144-145), only applicable to dynamic networks
- Multiplayer versions of the four interpersonal decision problems (ibid., pp. 238–266)

Dynamic networks were not included in the new model. Following the approach of systematically combining parameters, the additional structural dynamics created by the changing topology of the network would have to be incorporated into all the previous network topologies. Both this addition and the multiplayer versions of the game-theoretic games are beyond the scope of the model.

6.2 Quantitative Validation

Alexander provides quantitative results for a number of experiments. Given his foremost analytical approach, these are unfortunately mostly limited to simulations using bounded-degree networks and one experiment using lattices. The following describes the validation results for four different data sets, one for each interpersonal decision problem. Note that the initialization process might differ among the experiments. For the simulations of the new model, the initialization described in section 4.5 is used with an Initial Moral Mean of 0.5. Additionally, the generation of bounded-degree networks seems to differ between the two models. Alexander mentions that the networks used in his experiments are oftentimes disconnected for low degree intervals like $[1, 2]$ or $[1, 3]$ (ibid., p. 87). In contrast, the network generation used in the new model makes sure to never produce disconnected networks.

6.2.1 Prisoner's Dilemma

For the Prisoner's Dilemma, Alexander provides some data for bounded-degree networks with different degree intervals (ibid., pp. 86–87). The experiment involves running the simulation 10000 times using random initializations, counting the number of runs in which the population converges to a state in which all agents play *Defect*. The Prisoner's Dilemma was configured as $T = 1.0, R = 0.666, P = 0.333, S = 0.0$. Three different population sizes were used: 15, 30 and 60. It is not clear which learning rule was configured, thus *Imitate Best* was chosen, because it is the most common in Alexander's experiments. The comparison of results is shown in tables 6.1, 6.2 and 6.3 for the population sizes 15, 30 and 60, respectively. Note that the degree interval $[1, 2]$ was left out for the experiments of the new model, because it produces too many disconnected networks in practice (see section 5.2.2).

The results show that the general trends of the values match. As both d_{min} and d_{max} increase, the number of stable states of defection increases as well. Additionally, as the population size rises, the numbers decrease for low degrees and increase for high degrees. However, the individual numbers differ significantly for lower degrees. The results of the new model show a much smaller difference between low and high degrees (e.g. a difference of 901 for the new model vs. 2309 for Alexander, between degree intervals $[1, 3]$ and $[9, 10]$ for 15 agents). These differences shrink as we move towards higher degrees. Considering Alexander's remarks about networks for lower degree intervals oftentimes being disconnected, it is plausible that this is the main culprit of the differences. As the degrees increase, the number of disconnected networks in Alexander's model decreases. With fewer disconnected networks, the probability for a stable cooperating subgraph to evolve becomes smaller as well. This hypothesis can explain the differences the in results very well.

		d_{max}								
	d_{min}	2	3	4	5	6	7	8	9	10
Seil	1	—	9096	9513	9725	9788	9845	9859	9849	9857
	2	—	9490	9751	9804	9889	9906	9916	9885	9871
	3	—	—	9946	9941	9953	9951	9943	9947	9927
	4	—	—	—	9990	9993	9986	9978	9973	9971
	5	—	—	—	—	9993	9997	9991	9990	9983
	6	—	—	—	—	—	9998	9998	9995	9990
	7	—	—	—	—	—	—	9998	9998	9999
	8	—	—	—	—	—	—	—	9998	9997
	9	—	—	—	—	—	—	—	—	9997
Alexander	1	5185	7111	8062	8433	8686	8752	8720	8760	8692
	2	—	8410	8688	8812	8884	8871	8893	8884	8853
	3	—	—	9244	9192	9112	9093	9007	8963	8954
	4	—	—	—	9370	9354	9215	9199	9169	9114
	5	—	—	—	—	9363	9371	9312	9279	9197
	6	—	—	—	—	—	9369	9350	9329	9339
	7	—	—	—	—	—	—	9346	9396	9345
	8	—	—	—	—	—	—	—	9365	9409
	9	—	—	—	—	—	—	—	—	9420

Table 6.1: Comparison against Alexander (2007, p. 87): Number of runs out of 10000 converging to all agents playing *Defect* in the Prisoner's Dilemma, using bounded-degree networks with **15** agents.

		d_{max}								
	d_{min}	2	3	4	5	6	7	8	9	10
Seil	1	—	8437	9185	9601	9869	9927	9970	9991	9991
	2	—	9023	9561	9750	9906	9957	9982	9984	9992
	3	—	—	9892	9927	9961	9978	9990	9992	9995
	4	—	—	—	9989	9992	9994	9995	9998	9997
	5	—	—	—	—	10000	9999	9999	9998	10000
	6	—	—	—	—	—	9999	9999	9999	10000
	7	—	—	—	—	—	—	10000	10000	10000
	8	—	—	—	—	—	—	—	10000	10000
	9	—	—	—	—	—	—	—	—	10000
Alexander	1	4091	6003	7438	8455	8937	9199	9227	9262	9287
	2	—	7781	8557	8907	9253	9324	9361	9333	9368
	3	—	—	9442	9423	9496	9484	9475	9412	9448
	4	—	—	—	9627	9660	9581	9545	9502	9492
	5	—	—	—	—	9683	9667	9637	9601	9558
	6	—	—	—	—	—	9693	9686	9688	9642
	7	—	—	—	—	—	—	9673	9668	9671
	8	—	—	—	—	—	—	—	9660	9689
	9	—	—	—	—	—	—	—	—	9689

Table 6.2: Comparison against Alexander (2007, p. 87): Number of runs out of 10000 converging to all agents playing *Defect* in the Prisoner's Dilemma, using bounded-degree networks with **30** agents.

		d_{max}								
	d_{min}	2	3	4	5	6	7	8	9	10
Seil	1	—	7156	8581	9257	9728	9862	9964	9974	9996
	2	—	8154	9187	9479	9835	9912	9975	9988	9992
	3	—	—	9826	9807	9915	9937	9979	9992	9990
	4	—	—	—	9972	9982	9984	9992	9992	10000
	5	—	—	—	—	9999	9994	9998	9997	9999
	6	—	—	—	—	—	10000	10000	9999	10000
	7	—	—	—	—	—	—	10000	9999	10000
	8	—	—	—	—	—	—	—	10000	10000
	9	—	—	—	—	—	—	—	—	10000
Alexander	1	3076	4858	6441	7798	8790	9228	9482	9520	9621
	2	—	6850	8113	8689	9270	9446	9587	9598	9664
	3	—	—	9412	9319	9547	9617	9646	9702	9708
	4	—	—	—	9760	9788	9777	9781	9751	9745
	5	—	—	—	—	9828	9826	9796	9793	9779
	6	—	—	—	—	—	9851	9836	9831	9835
	7	—	—	—	—	—	—	9806	9827	9831
	8	—	—	—	—	—	—	—	9842	9817
	9	—	—	—	—	—	—	—	—	9858

Table 6.3: Comparison against Alexander (2007, p. 87): Number of runs out of 10000 converging to all agents playing *Defect* in the Prisoner's Dilemma, using bounded-degree networks with **60** agents.

6.2.2 Stag Hunt

For the Stag Hunt, there is data for bounded-degree networks with degree interval [2, 4] (Alexander 2007, pp. 137–138). For this experiment, the proportion of agents using the strategy *Stag* in the risk-dominant Stag Hunt was recorded at the stable state of the population. The simulations are executed 10000 times each for the learning distances 1 and 2, using random initializations. The configured learning rule is *Imitate Best*, the population size is 40. The data is shown in table 6.4.

	Number of simulations			
	Learning Distance = 1		Learning Distance = 2	
Proportion p of *Stag* players	Seil	Alexander	Seil	Alexander
$p = 1$	68	1517	8139	5818
$0.9 \leq p < 1.0$	237	909	67	138
$0.8 \leq p < 0.9$	332	554	6	91
$0.7 \leq p < 0.8$	522	426	4	66
$0.6 \leq p < 0.7$	679	389	0	70
$0.5 \leq p < 0.6$	932	351	3	40
$0.4 \leq p < 0.5$	1144	418	3	50
$0.3 \leq p < 0.4$	1392	456	14	35
$0.2 \leq p < 0.3$	1487	571	16	59
$0.1 \leq p < 0.2$	1612	851	21	40
$0.0 < p < 0.1$	115	76	6	7
$p = 0$	1480	3482	1721	3586

Table 6.4: Comparison against Alexander (2007, pp. 137–138): Number of runs out of 10000 converging to the specified outcomes in the Stag Hunt, using bounded-degree networks with degree interval [2, 4].

The results are matching in the trend observed by increasing the learning distance. The strategies reported in the stable states are much more varied for a learning distance of 1, whereas they are accumulated at the extremes for a learning distance of 2. The distribution of strategies is very different between the two models for a learning distance of 1, however. While Alexander's results simply show a smaller extent of the accumulation at the extremes, the result for the new model are mostly centered around $0.2 \leq p < 0.3$. It is unclear what causes this stark difference. One possibility is the differing initializations. Alexander mentions that for this experiment, the initial strategies are "selected according to a randomly chosen distribution, in addition to being randomly assigned to individuals" (ibid., p. 137). Conceptually, this sounds similar to the approach described in section 4.5. What the available distributions are, though, and how exactly they are chosen, is unclear and may have a strong influence on the results.

6.2.3 Bargaining Subgame

For the Bargaining (Sub-)Game, Alexander reports an experiment using two-dimensional lattices which counts the number of times the population converges to a polymorphic stable state (Alexander 2007, pp. 170–173). The polymorphism *4-6*, for example, specifies the state of the population in which every agent plays either *Demand 4* or *Demand 6*. The model is configured with the learning rules *Imitate Probability*, *Imitate Best* and *Imitate Average* as well as three different neighborhoods: von Neumann with distance 1 (N4), Moore with distance 1 (M8) and Moore with distance 2 (M24). Again, the simulations are run 10000 times for each configuration with random initializations. It is not clear how many agents Alexander chose for his experiments, so a 32x32 lattice with wrap-around was configured for the simulations of the new model. The comparison of results is shown in table 6.5. For this experiment, the full Bargaining Game was added next to the Bargaining Subgame, in order to see whether the results differ significantly among the two variations.

The results between the two models are very similar. Interestingly, using the Bargaining Subgame instead of the full Bargaining Game seems to make almost no difference. Alexander's data is slightly more spread among the polymorphisms *5* and *4-6* as well as other final states, while the new model results for the full Bargaining Game do not show a single simulation which did not converge to all agents playing *Demand 5*. These small differences could be explained by a variety of potential differences in configurations, including initialization, population size and wrap-around of the lattice.

			Polymorphism						
	Nbhd.	Learning Rule	0–10	1–9	2–8	3–7	4–6	5	Other
Seil *subgame*	N4	Imitate Probability	0	0	0	0	0	10000	0
	N4	Imitate Best	0	0	0	0	0	10000	0
	N4	Imitate Average	0	0	0	0	0	10000	0
	M8	Imitate Probability	0	0	0	0	0	10000	0
	M8	Imitate Best	0	0	0	0	0	10000	0
	M8	Imitate Average	0	0	0	0	0	10000	0
	M24	Imitate Probability	0	0	0	0	0	10000	0
	M24	Imitate Best	0	0	0	0	0	9998	2
	M24	Imitate Average	0	0	0	0	0	9998	2
Seil *full game*	N4	Imitate Probability	0	0	0	0	0	10000	0
	N4	Imitate Best	0	0	0	0	0	10000	0
	N4	Imitate Average	0	0	0	0	0	10000	0
	M8	Imitate Probability	0	0	0	0	0	10000	0
	M8	Imitate Best	0	0	0	0	0	10000	0
	M8	Imitate Average	0	0	0	0	0	10000	0
	M24	Imitate Probability	0	0	0	0	0	10000	0
	M24	Imitate Best	0	0	0	0	0	10000	0
	M24	Imitate Average	0	0	0	0	0	10000	0
Alexander *full game*	N4	Imitate Probability	0	0	0	0	29	9970	1
	N4	Imitate Best	0	0	0	0	26	9966	8
	N4	Imitate Average	0	0	0	0	13	9984	3
	M8	Imitate Probability	0	0	0	0	26	9973	1
	M8	Imitate Best	0	0	0	0	26	9908	66
	M8	Imitate Average	0	0	0	0	24	9970	6
	M24	Imitate Probability	0	0	0	8	110	9879	3
	M24	Imitate Best	0	0	0	21	220	9721	38
	M24	Imitate Average	0	0	0	0	62	9934	4

Table 6.5: Comparison against Alexander (2007, p. 172): Number of runs out of 10000 converging to the specified polymorphisms in the Bargaining (Sub-)Game, using two-dimensional lattices.

6.2.4 Ultimatum Subgame

For the Ultimatum Subgame, Alexander provides data for bounded-degree networks using different degree intervals (Alexander 2007, pp. 231–233). Here, two specific polymorphic stable states of the population were recorded. The model is configured with a population size of 50, and *Imitate Best* as the learning rule. The simulations were repeated 1000 times each for mutation disabled and enabled (using a mutation rate of 0.02), both times with randomly initialized populations. Tables 6.6 and 6.7 show the data for mutation disabled and enabled, respectively.

The results share the trend that the number of immoral polymorphisms (agents playing either *S1 (Gamesman)* or *S4 (Mad Dog)*) is generally higher when mutation is enabled. Apart from this, there are some interesting differences, though. In Alexander's experiments, the number of moral polymorphisms (agents playing either *S7 (Fairman)* or *S5 (Easy Rider)*) generally decreases when mutation is enabled. In the new experiments, there is a slight increase especially for smaller degrees. One possible explanation for this is the following: There are a few additional cases where it would be possible, given the right mutations, for moral strategies to spread as long as the connectivity of the network is low. The abundance of disconnected networks when using low degrees in Alexander's model prevents exactly this spread, resulting in lower numbers of such polymorphisms. A more significant difference between the two models is that in Alexander's case, different degree intervals have little influence on the results. For the new model simulations, the degrees have a strong influence as long as mutation is disabled (e.g. a result of *61, 7* vs. *917, 44* for degree intervals $[2, 3]$ and $[9, 10]$). It is unclear where this difference comes from. It is possible that for low degrees, the initialization mostly determines the outcome of the simulations. When activating mutations, this influence is obviously reduced. If Alexander uses a different initialization process, this could explain the differences in the results.

		d_{max}							
	d_{min}	3	4	5	6	7	8	9	10
S.	2	61, 7	184, 20	358, 43	500, 81	627, 94	690, 126	715, 128	750, 120
	3	—	221, 18	369, 47	534, 53	626, 97	673, 107	743, 101	758, 126
	4	—	—	352, 36	535, 45	645, 71	701, 87	717, 95	747, 110
	5	—	—	—	490, 36	659, 43	698, 72	747, 68	775, 95
	6	—	—	—	—	571, 40	662, 59	727, 65	810, 67
	7	—	—	—	—	—	609, 31	698, 38	855, 52
	8	—	—	—	—	—	—	580, 40	882, 44
	9	—	—	—	—	—	—	—	917, 44
A.	2	525, 64	605, 107	711, 140	734, 163	734, 183	721, 216	715, 215	708, 228
	3	—	767, 131	772, 148	765, 150	736, 197	755, 192	731, 211	725, 225
	4	—	—	829, 118	796, 163	778, 162	752, 191	762, 190	737, 197
	5	—	—	—	797, 143	777, 168	770, 178	773, 171	751, 180
	6	—	—	—	—	795, 134	773, 155	776, 166	728, 196
	7	—	—	—	—	—	793, 153	796, 145	770, 152
	8	—	—	—	—	—	—	791, 123	775, 140
	9	—	—	—	—	—	—	—	764, 144

Table 6.6: Comparison against Alexander (2007, p. 232): Number of runs out of 1000 converging to different polymorphisms, using bounded-degree networks with 50 agents, with mutation **disabled**. Each entry g, f corresponds to the number of runs g which converged to all agents playing either *S1 (Gamesman)* or *S4 (Mad Dog)*, and the number of runs f which converged to all agents playing either *S7 (Fairman)* or *S5 (Easy Rider)*.

		d_{max}							
	d_{min}	3	4	5	6	7	8	9	10
S.	2	664, 20	888, 63	900, 88	875, 124	875, 125	873, 127	875, 125	855, 145
	3	—	929, 55	929, 68	918, 79	880, 119	893, 107	890, 110	878, 122
	4	—	—	934, 66	932, 68	931, 69	897, 101	907, 93	890, 110
	5	—	—	—	959, 40	947, 53	950, 50	941, 59	906, 92
	6	—	—	—	—	960, 40	959, 41	935, 64	939, 61
	7	—	—	—	—	—	968, 32	956, 44	956, 42
	8	—	—	—	—	—	—	972, 28	964, 36
	9	—	—	—	—	—	—	—	966, 33
A.	2	695, 1	743, 4	769, 16	780, 20	828, 31	818, 35	833, 34	854, 42
	3	—	819, 3	859, 8	820, 16	835, 25	840, 41	845, 43	855, 36
	4	—	—	868, 4	844, 23	856, 15	854, 25	876, 31	857, 41
	5	—	—	—	882, 8	888, 13	885, 14	864, 21	873, 26
	6	—	—	—	—	882, 4	885, 14	876, 16	891, 16
	7	—	—	—	—	—	904, 5	906, 9	907, 10
	8	—	—	—	—	—	—	908, 4	912, 12
	9	—	—	—	—	—	—	—	914, 7

Table 6.7: Comparison against Alexander (2007, p. 232): Number of runs out of 1000 converging to different polymorphisms, using bounded-degree networks with 50 agents, with mutation **enabled**. Each entry g, f corresponds to the number of runs g which converged to all agents playing either *S1 (Gamesman)* or *S4 (Mad Dog)*, and the number of runs f which converged to all agents playing either *S7 (Fairman)* or *S5 (Easy Rider)*.

Chapter 7

Sensitivity Analysis

Sensitivity analysis can in general be described as "the study of how the uncertainty in the output of a model (numerical or otherwise) can be apportioned to different sources of uncertainty in the model input" (Saltelli et al. 2004, p. 45). In other words, we want to find out which parameters and parameter combinations influence the result of the simulations in what way. In doing so, we can figure out the most relevant components of the system that is being modelled, as well as uncover parts of the model which do not contribute to the behaviour of the system at all. This chapter first introduces the specific methodology which is used for the sensitivity analysis in section 7.1. The experimental setup and the choice of the sampled parameter values are presented in section 7.2. The results of the analysis are then laid out in section 7.3.

7.1 Methodology

There are many different methods one can choose to carry out a sensitivity analysis (Broeke, Voorn, and Ligtenberg 2016). Oftentimes, there is a trade-off between simplicity, generality and fast execution on the one side, and complexity, precision and high computational demands on the other. Thiele, Kurth, and Grimm (2014, sec. 3.54) recommend starting the inquiry into the model with a simple method, and then use these results to selectively employ more sophisticated methods. The sensitivity analysis in this paper thus takes the first step and uses the rather simple *one-factor-at-a-time* (OFAT) method (Broeke, Voorn, and Ligtenberg 2016, secs. 3.3-3.4) to provide a better understanding of the model mechanisms.

The OFAT method attempts to highlight the relationship between each single parameter of the model and the simulation results. It works as follows. First, we need to select a base value for each parameter, called the *nominal value*. All parameters set to their nominal values provide the *nominal configuration* of the model. We also need to define a number of sampled values for each parameter we want to analyze. Based on the nominal configuration, we then apply the sampled values of one parameter at a time, each time keeping all the other parameters at their nominal values. The simulations are executed for all sampled values of every parameter. For example, if the model contains the parameters A, B and C, we can define their nominal values A_n, B_n and C_n as well as their sampled values $\{A_1, A_2\}$, $\{B_1, B_2, B_3\}$ and $\{C_1, C_2\}$. We are then

going to execute the simulations for the following configurations:

- $(\boldsymbol{A_1}, B_n, C_n)$
- $(\boldsymbol{A_2}, B_n, C_n)$
- $(A_n, \boldsymbol{B_1}, C_n)$
- $(A_n, \boldsymbol{B_2}, C_n)$
- $(A_n, \boldsymbol{B_3}, C_n)$
- $(A_n, B_n, \boldsymbol{C_1})$
- $(A_n, B_n, \boldsymbol{C_2})$

In using this method, we approximate the execution of all possible sampled parameter value combinations by only looking at one parameter at a time. When using n different parameters and m different values per parameter, this reduces the runtime complexity of the approach from $\mathcal{O}(n^m)$ to $\mathcal{O}(n \cdot m)$. The downside of this method is that it disregards the influence of combinations of parameters on the output. If parameters A and B each have little influence on the results on their own, but the combination of certain values of A and B does have a strong influence, then this piece of information cannot be uncovered using the OFAT method. Still, it provides a very useful first step into understanding the behaviour of the model.

7.2 Experimental Setup

In order to use the OFAT method described in the last section, the general configuration of the model needs to be specified, as well as the the nominal and sampled parameter values. For the simulations of the sensitivity analysis, Homogeneity Detection and Cycle Detection are both enabled, except for when *Imitate Probability* is used as the learning rule, or when mutation is enabled. The Max Generation Count is set to 1000, based on manual experimentation which shows that most simulations converge after fewer than 100 generations. The Seed for the PRNG is set to 19561831. Given there are stochastic components in the initialization, as well as the mutation and possibly learning processes, each parameter configuration needs to be executed an adequate number of times, in order to account for variance among the results. There are many approaches to deciding which number of repetitions to choose (Thiele, Kurth, and Grimm 2014, sec 4.4). In this case, the number was chosen based on the available time and computing resources available. Therefore, each configuration is executed 100 times. The exception to this are the simulations using the fully connected network. They were limited to 10 runs, because the sub-optimal implementation of these networks turned out to slow down the simulations significantly.

The nominal and sampled values of the parameters are listed in table 7.1. In general, the approach was to choose at least three sampled values for each parameter, in order to be able to find potential non-linearities in the influence of a parameter on the simulation results. In the case of parameters for which the scale of possible values is not restricted in one direction, five values were chosen to achieve the same goal. For the Neighborhood and Wrap Around parameters, the same nominal and sampled values were chosen for each lattice and its small-world counterpart. This makes it possible to identify the difference between the two topologies by controlling for all other parameters. The lowest sampled value for the Neighborhood parameter of the one-dimensional lattice is therefore 2, because using a value of 1 for the one-dimensional small-world networks results in too many disconnected networks during the network generation. The Degree Interval parameter contains values that are supposed to show a progression in both absolute size of the degrees (e.g. $[2, 3]$ vs. $[8, 9]$) as well as in relative difference of the degrees (e.g. $[5, 6]$ vs. $[3, 8]$). The values for the Edge Probability parameter of the fully random network are chosen based on experimentation, while providing a linear increase in value.

Network/Game	Parameter	Sampled Values	Nominal
—	Population Size	100, 1000, 10000	1000
	Initial Moral Mean	0.25, 0.5, 0.75	0.5
	Learning Rule	IBest, IProb, IAvg, BestR	IBest
	Mutation Probability	0.0, 0.001, 0.01, 0.1	0.0
	Mutation Distance	0, 1, 2	0
Prisoner's Dilemma	Cooperation Incentive	0.2, 1.0, 5.0	1.0
	Defection Incentive	0.2, 1.0, 5.0	1.0
Stag Hunt	Risk Dominance	true, false	true
All Except F. C.	Learning Distance	1, 2, 3	1
Lattice 1D	Neighborhood	2, 3, 4	2
	Wrap Around	true, false	true
Lattice 2D	Neighborhood	N4, N12, N24, M8, M24, M48	M8
	Wrap Around	true, false	true
Small-World 1D	Neighborhood	2, 3, 4	2
	Wrap Around	true, false	true
	Beta	0.03, 0.07, 0.125, 0.18, 0.28	0.125
Small-World 2D	Neighborhood	N4, N12, N24, M8, M24, M48	M8
	Wrap Around	true, false	true
	Beta	0.008, 0.04, 0.09, 0.17, 0.275	0.09
Bounded-Degree	Degree Interval	$[2, 3]$, $[5, 6]$, $[8, 9]$, $[4, 7]$, $[3, 8]$	$[4, 7]$
Fully Random	Edge Probability	0.004, 0.008, 0.012	0.008

Table 7.1: Sampled values and nominal values used for the parameters of the sensitivity analysis.

In choosing the values for the Beta parameter of the small-world networks, the ω measure described by Qawi et al. (2011) was used. This measure quantifies the small-world property of graphs. Concretely, it compares the characteristic path length and the clustering coefficient of the graph under investigation to equivalent lattices and random graphs. The measure is defined in equation 7.1. The resulting ω values lie in the interval $[-1, 1]$, regardless of the size of the network. A value close to -1 describes a perfectly regular lattice, while a value close to 1 describes a completely random graph. Values close to 0 maximize the small-world properties of the graph. The approach taken was thus to find suitable Beta values which approximate corresponding ω values of -0.5, -0.25, 0, 0.25 and 0.5, in order to provide a progression in both directions. These final values are listed in table 7.2. This was done by generating small-world networks with 1000 agents and calculating their ω measures. Given the random generation of the small-world networks, the ω values were averaged over 100 runs. For the one-dimensional small-world network, the equivalent lattice was chosen to be a one-dimensional lattice with 1000 agents, a Neighborhood of 2 and Wrap Around set to true. The equivalent random graph was chosen to be a bounded-degree network with 1000 agents and a Degree Interval of $[4, 4]$. This results in the values $C_{lattice} = 0.5$ and $L_{random} \approx 5.636$ (the latter averaged over 100 generations). For the two-dimensional small-world network, the lattice was a two-dimensional lattice with 1024 agents (32x32), the Neighborhood M8 and Wrap Around set so true. The random graph was a bounded-degree network with 1000 agents and a Degree Interval of $[8, 8]$. This delivers the values $C_{lattice} \approx 0.429$ and $L_{random} \approx 3.610$ (the latter again averaged over 100 generations).

$$
\begin{aligned}
L &= \text{characteristic path length of small-world graph} \\
L_{random} &= \text{characteristic path length of equivalent random graph} \\
\\
C &= \text{clustering coefficient of small-world graph} \\
C_{lattice} &= \text{clustering coefficient of equivalent lattice} \\
\\
\omega &= \frac{L_{random}}{L} - \frac{C}{C_{lattice}}
\end{aligned}
\tag{7.1}
$$

There are a few special cases when carrying out the simulations of the sensitivity analysis. The Population Size of 10000 was disabled for both the fully connected network and the fully random network, because the simulations exceeded the available memory of the system being used in both cases. Additionally, the Population Size of 100 was disabled for fully random networks, because when using the specified values for Edge Probability, the network generation produced too many disconnected networks for this small amount of agents. Finally, when using the sampled values of the Mutation Distance parameter, the nominal value of the Mutation Probability parameter was set to 0.01 instead of 0.0. For the fully connected network, the Mutation Distance is always set to 0.

Network	β	ω
Small-World 1D	0.03	-0.53
	0.07	-0.27
	0.125	0.02
	0.18	0.23
	0.28	0.51
Small-World 2D	0.008	-0.50
	0.04	-0.24
	0.09	-0.01
	0.17	0.27
	0.275	0.50

Table 7.2: Average ω measures (rounded to two decimal places) over 100 network generations, corresponding to different β values for one- and two-dimensional small-world networks.

7.3 Results

The complete collection of result graphs as well as the raw result data of the sensitivity analysis can be found at `https://github.com/srseil/Seom/tree/v1.0.0` *as well as at* `https://pms.ifi.lmu.de/publications/diplomarbeiten/Stefan.Seil/dataset.zip`.

In this section, the results of the sensitivity analysis are presented in a descriptive way, while staying within the context of the model. Chapter 8 discusses some of the interesting results in more detail, focusing on the implications for the Structural Evolution of Morality (Alexander 2007). Note that the presentation of the results mostly focuses on the morality measure. This is done because the stability results are almost always accumulated at the extremes. When a stochastic component is used, the stability result is always 0, because all generations have to be considered. For most other cases, the simulations converge to a small cycle, leading to very high stability results. Only in a few specific cases does the stability measure report interesting results.

The relationship between parameter values and result measures is sometimes visualized in graphs. These graphs contain three main components: a box plot, a violin plot surrounding it, and a curve fitted through the mean result values. As is common, the box plot displays the quartiles of the result data, whereby the dot inside the box visualizes the mean value, and the bar inside the box visualizes the median value. Since box plots cannot properly show the probability density of different values, the graphs are supplemented with violin plots. The wider the body of the violin plot is at a particular value, the higher the density of data points is at this value. Note that the box plots and violin plots oftentimes degenerate to very small boxes, or even singular points, when the variance of the result values is very low. Additionally, the graphs include a polynomial curve fitted through the mean points of the box plots, calculated with local regression (*locally estimated scatterplot smoothing*). This is not an attempt to provide a plausible fit through the sampled values, given that there are too few sampled values for most parameters. Rather, the purpose is to more clearly show the general progression of mean points

over multiple parameter values, in order to ease the visual understanding of the high informational density in the graphs. Finally, the nominal value for each parameter is highlighted with a dotted line.

7.3.1 General

There are a number of results which apply to the majority of games and network topologies. Generally, each of the four games has a baseline result that dominates over the majority of parametrizations. Certain configurations can then diverge from this result. These baseline results are visualized at the beginning of the respective subsections 7.3.3 to 7.3.6. The Stag Hunt and the Bargaining Subgame generally converge to close to maximum morality, while the Prisoner's Dilemma and Ultimatum Subgame typically result in a morality measure close to zero. The stability measure is also close to 1.0 in most of these cases. The Ultimatum Subgame is the only game where both measures consistently vary a bit more from their extremes. For all games except the Bargaining Subgame, the baseline result also conforms to the nominal configuration on the fully connected network (see section 7.3.2). Across all games, the one-dimensional lattice is the topology which consistently shows the highest results for morality. One-dimensional small world networks oftentimes increase the variance of the results significantly. The two-dimensional small-world networks, however, do not typically exhibit this behaviour. The fully connected network is, as expected, oftentimes an odd-one-out, considering it misses any structural constraints. The other topologies (two-dimensional lattices, two-dimensional small-world networks, bounded-degree networks and fully random networks) tend to have very similar results for most of the parameters.

Initial Moral Mean

For all games except the Prisoner's Dilemma, increasing the Initial Moral Mean has an effect towards increasing morality. How strong this effect is depends on the game and network topology. Figure 7.3 shows examples of this effect. In contrast, changing the Population Size doesn't have any effect on the results, except for three cases: the Stag Hunt on a fully connected network (see section 7.3.2), as well as one-dimensional small-world networks in the Bargaining Subgame (see section 7.3.5) and the Ultimatum Subgame (see section 7.3.6).

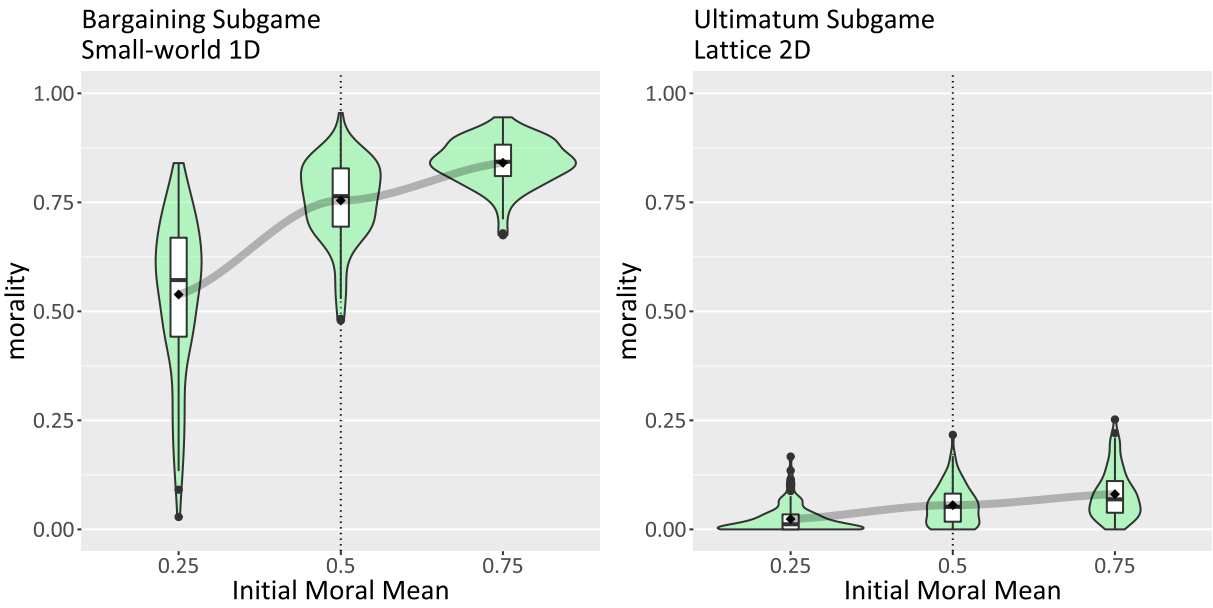

Figure 7.3: Examples for positive influence of Initial Moral Mean on morality.

Mutation Distance & Mutation Probability

Increasing the values for the Mutation Distance proportionally moves the results away from the baseline. Presumably, this is because the mutation process intermittently creates clusters of agents which do not conform to the baseline strategy, which are quickly overtaken by the rest of the population again. Because the cycle used for the morality and stability measures consists of all 1000 generations when mutation is enabled, these intermittent changes still change the results of the simulation.

Increasing the Mutation Probability has a similar effect, although it typically takes a high value (0.1) to make any significant impact on the results. This seems straightforward: Clusters of mutated agents can survive longer than individual mutants, because their neighborhood is filled with individuals playing the same strategy, which can reinforce their choice of strategy for some time. Examples for both parameters are shown in figure 7.4. The exceptions to this behaviour are the Stag Hunt on a fully connected network (see section 7.3.2), and the Ultimatum Subgame generally, where the effects are reversed (see section 7.3.6).

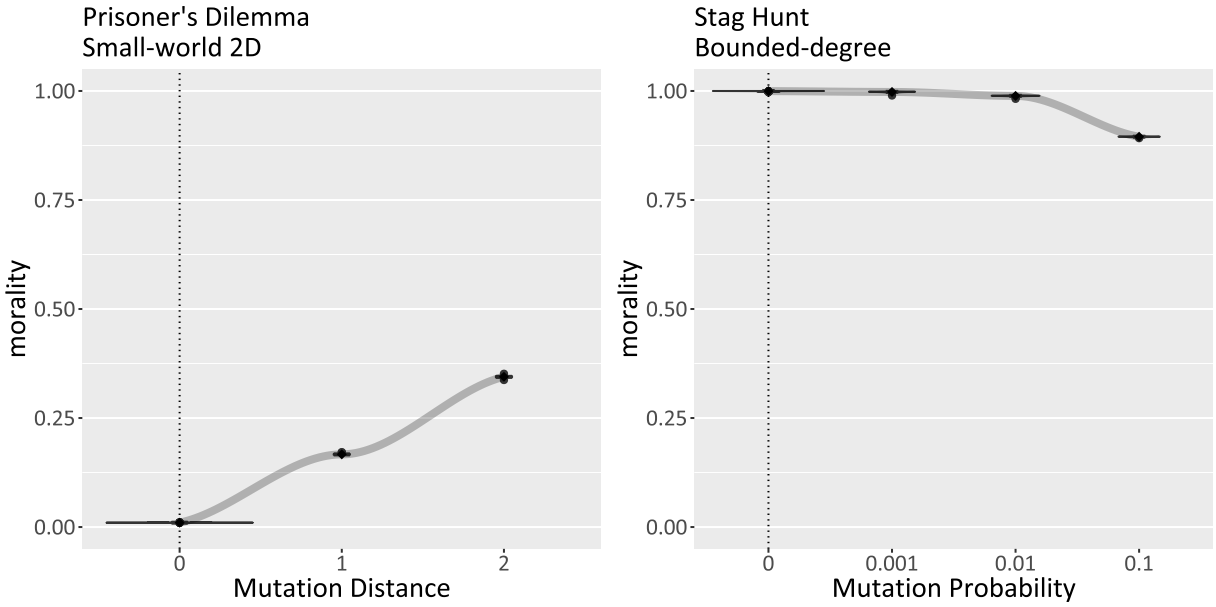

Figure 7.4: Examples of Mutation Distance and Mutation Probability moving morality results away from the baseline.

Learning Distance

The Learning Distance parameter does not have much impact on the results at all, except for a few specific cases shown in figure 7.5. One reason for this might be that most of the games under investigation do not have many strategies available to them. Thus, an agent is likely to find an instance of each of the available strategies within a smaller learning neighborhood. Considering one exception to this occurs in the Ultimatum Subgame, which has more strategies than the other three games, this seems like a plausible explanation.

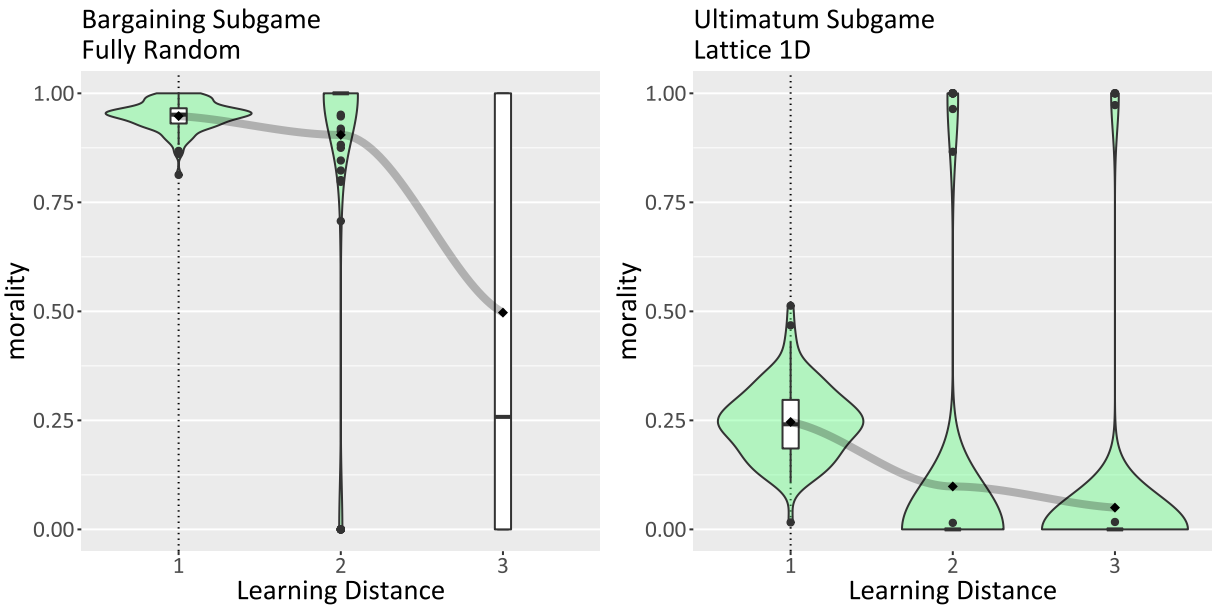

Figure 7.5: Examples of rare significant influence of Learning Distance on morality.

Learning Rule

Among the learning rules, *Imitate Best* and *Imitate Probability* lead to very similar results across the board. In some cases, *Imitate Average* creates a negative influence on the morality measure. Most interestingly, *Best Response* typically has a very different influence on the results compared to the other rules. This is to be expected, because it models more sophisticated reasoning capabilities than the other three learning rules. The effects of *Best Response* on the morality measure are mostly negative, though. Figure 7.6 shows an example of this on two-dimensional lattices.

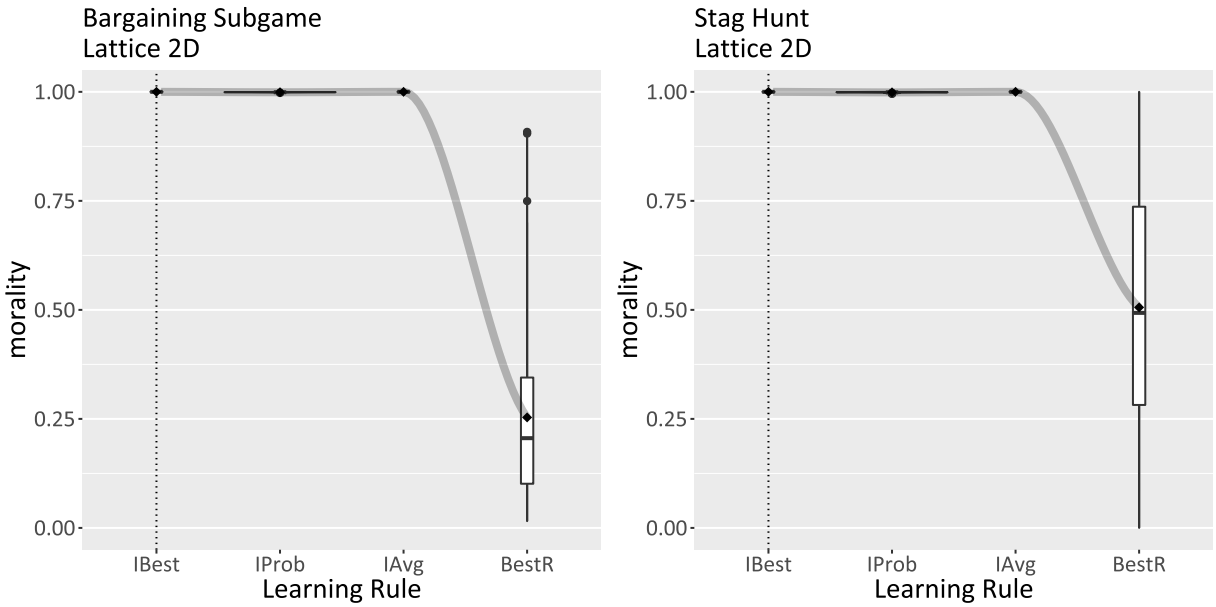

Figure 7.6: Examples of negative influence of *Best Response* learning rule on morality.

Neighborhood & Wrap Around

The Neighborhood of the lattices has almost no effect on the results, except for a small decrease in the morality and stability measures in the Ultimatum Subgame. For the small-world networks, very small neighborhoods move the morality results away from the baseline a bit. We can assume that the same reason as with the Learning Distance parameter applies: Given the low number of strategies, the size of an agent's neighborhood does not matter much, unless we restrict it to a very small size. Figure 7.7 shows examples of both behaviours. In contrast, there is not a single case in which the Wrap Around parameter makes any significant difference for the results. Apparently, it is not relevant whether there are less-connected corners and edges on a lattice, as long as the population is large enough.

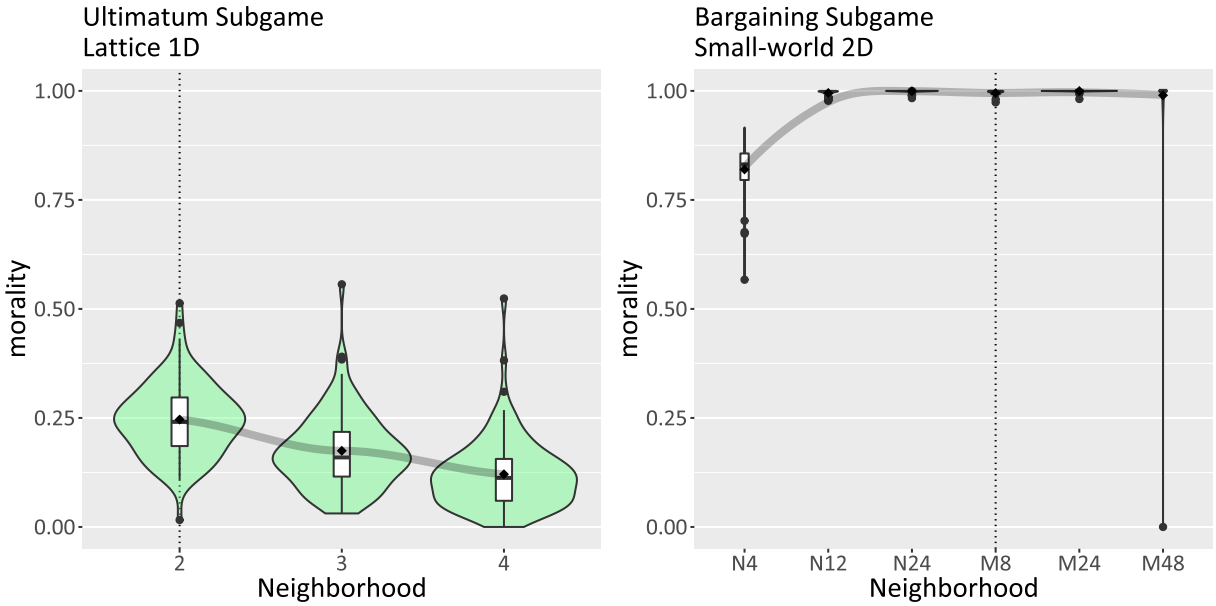

Figure 7.7: Examples of impact of Neighborhood on morality on lattices and small-world networks.

Beta

The Beta parameter specifying the small-world property of small-world networks shows interesting results. Except for the Prisoner's Dilemma (where there is no effect), one-dimensional small-world networks move the morality results away from baseline. For the Bargaining Subgame, and to a lesser extent for the Stag Hunt, this happens proportionally to the magnitude of the small-world properties. Concretely, the β values which correspond to ω values close to 0 (near maximum small-world properties) show the strongest effect, which tapers off when moving towards higher or lower ω values (closer to random networks and lattices, respectively). The effect on the results thus forms a parable, with the minimum value at the maximum small-world property of the network. Figure 7.8 shows an example of this. Outside of the Bargaining Subgame, however, the stability results are mostly flat. Additionally, for the two-dimensional small-world networks, the parameter has almost no effect on the result measures regardless of which game is chosen. It is unclear what causes this behaviour, especially considering that it only occurs for the one-dimensional small-world networks. One possible explanation is that as long as an agent's neighborhood is generally small (as is the fact on one-dimensional small-world networks), multiple agents connected by rewired edges can survive using a different strategy than the rest of the population.

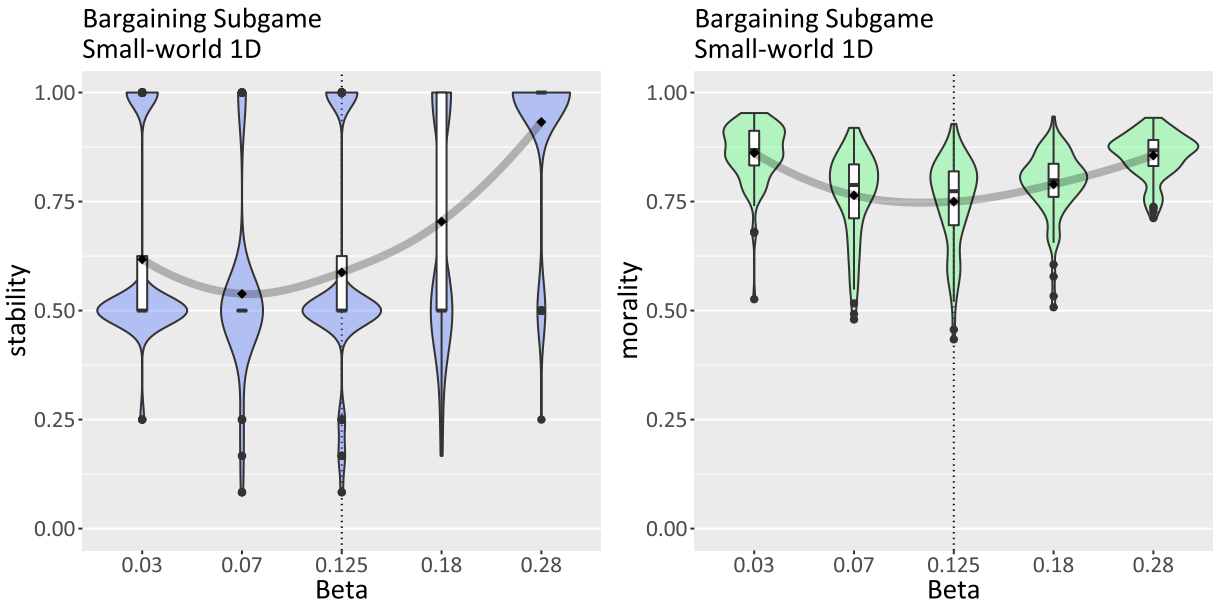

Figure 7.8: Example of influence of Beta on stability and morality on one-dimensional small-world networks.

Degree Interval

For bounded-degree networks, the choice of the Degree Interval does not seem to make a difference, with the exception of one of the sampled values. Across all games, the interval [2, 3] consistently moves the result a noticeable amount away from the baseline result. An example of this is shown in figure 7.9. To a lesser extent, this is also true for the Edge Probability parameter, where only the value 0.004 has a noticeable influence on the result measures. There is one explanation that comes to mind: While the networks generated by the implementation are guaranteed to be connected, it can happen that the graph contains multiple connected components which are only connected to each other via one single edge. One can imagine that inside one of these components, a strategy different from the baseline survives, because the single edge to the outside world is not enough for the cluster to be overtaken.

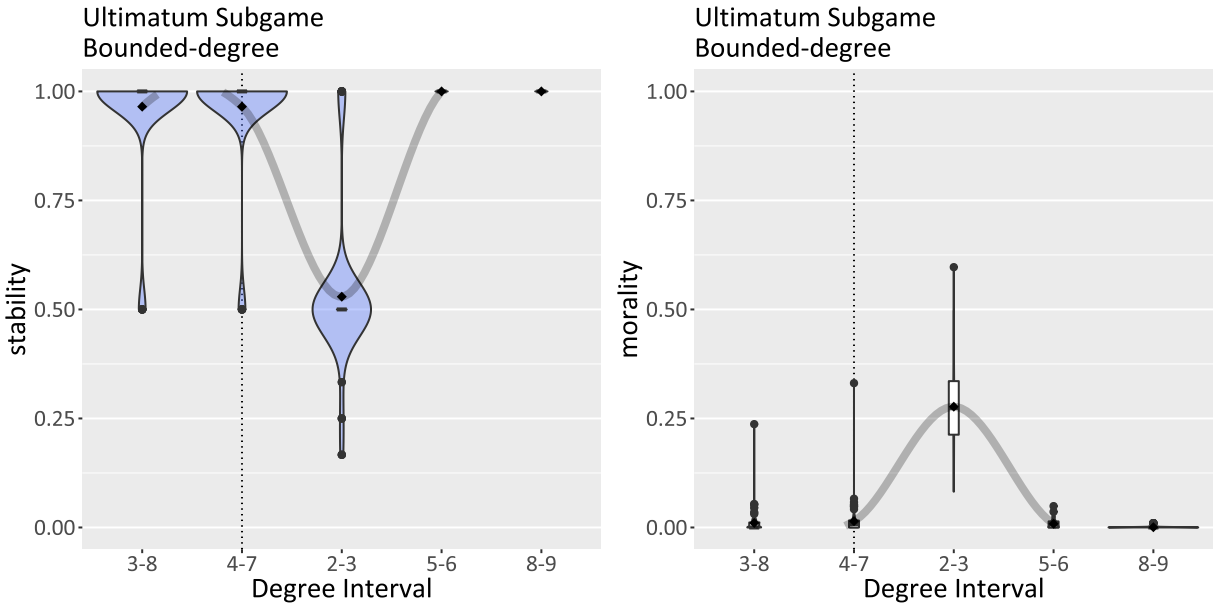

Figure 7.9: Example of influence of Degree Interval on stability and morality.

7.3.2 Fully Connected Network

As alluded to in chapter 4, the fully connected network topology was meant to be both a control topology to analyze the influence of different structural constraints on the evolutionary dynamics, as well as an attempt to imitate the replicator dynamics in an agent-based model. This section analyzes how the fully connected network fares against the replicator dynamics.

Prisoner's Dilemma

For the Prisoner's Dilemma, the results of the fully connected network perfectly match the replicator dynamics. No parameter has any influence on these results. The results of both models are shown in figures 7.10 and 7.11. In this case, mimicking the replicator dynamics seems easy, as the evolutionary dynamics always move the population towards a state of all individuals playing defect.

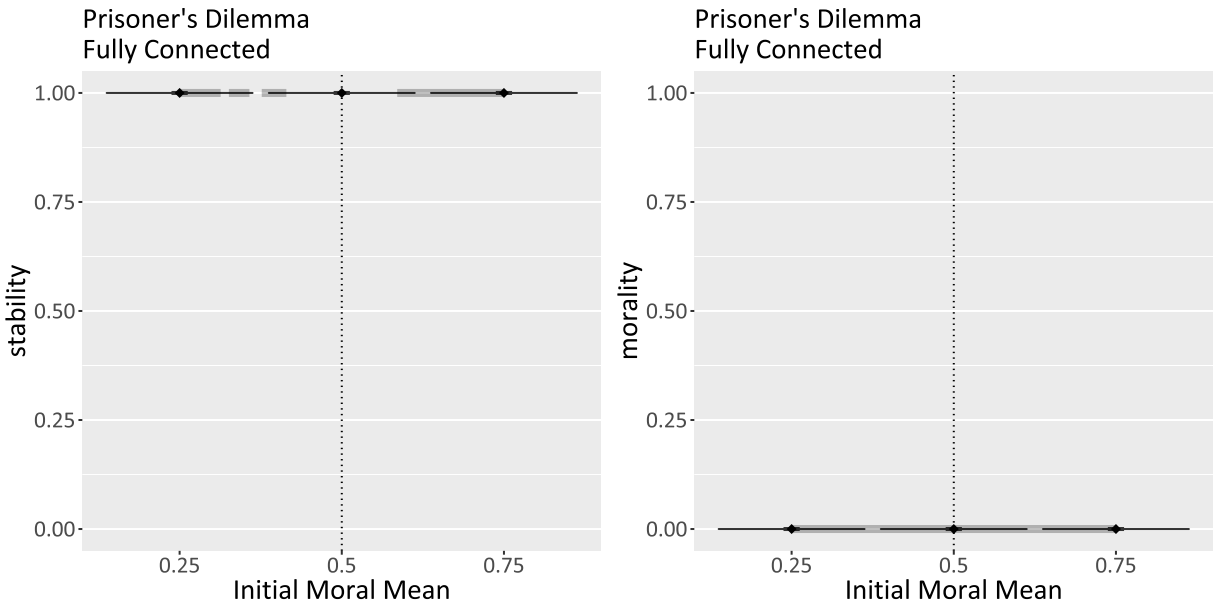

Figure 7.10: Results for the Prisoner's Dilemma on fully connected networks.

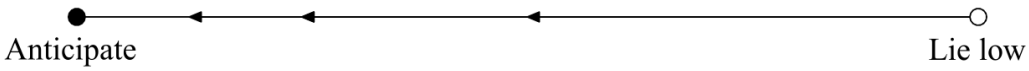

Figure 7.11: Results for the Prisoner's Dilemma using the replicator dynamics (Alexander 2007, p. 57). *Lie low* corresponds to *Cooperate*, and *Anticipate* corresponds to *Defect*. If there is at least one individual using the strategy *Defect*, then the population is moved towards the state in which all individuals play *Defect*.

Stag Hunt

In the Stag Hunt on the fully connected network, the Initial Moral Mean completely determines the morality result (shown in figure 7.12). The results do not change with different learning rules, or when using the non-risk dominant version for the *Stag* strategy of the game. The replicator dynamics tell a similar story, in that the state which the population converges to only depends on the initial distribution of strategies (Alexander 2007, p. 112). However, how many *Stag* players need to be initially present, in order to let the population converge to all individuals playing *Stag*, is somewhat different. For the payoff matrix used in the agent-based model, the tipping point is calculated as $p = \frac{1}{3-2+1} = 0.5$. If more than half of the population starts out as *Stag* players, then the population converges to the Stag equilibrium.

When looking at the results from the agent-based model, the tipping point seems to be higher. If one imagines that the relationship between the parameter and the results is linear, then the tipping point for the agent-based model would have to be somewhere between 0.5 and 1.0. Note that the variance of the results simply comes from the fact that the initialization doesn't use fixed probabilities for the strategies, but samples them from a probability density function characterized by the Initial Moral Mean parameter. The violin plot also paints a deceiving picture: The results are always at the extremes 0.0 or 1.0, and never in between. When analyzing the Population Size parameter, we can see that reducing its value by one order of magnitude results in a slightly lower tipping point. Due to the sub-optimal implementation of the fully connected networks, no population size larger than 1000 could be chosen for the experiments. Based on these preliminary results it seems plausible, though, that the tipping point would converge towards 0.5 when increasing the population size further.

Figure 7.12: Results for the Stag Hunt on fully connected networks, influenced by the Initial Moral Mean and the Population Size.

Bargaining Subgame

The Bargaining Subgame shows interesting results on the fully connected network. In absence of any stochastic effects, the simulation always results in minimum morality. When using *Imitate Probability* as the learning rule, though, or when activating mutation, it achieves very high morality results. The results for both the agent-based model and the replicator dynamics are shown in figure 7.13. In the replicator dynamics, the population can converge to one of two different equilibria (all individuals playing Demand 5, or some individuals playing *Demand 4* and some playing *Demand 6*) depending on the initial state of the population. When looking at the simplex diagram for the replicator dynamics in figure 7.13, though, we see that an initialization which divides the initial strategies approximately equally among the population (i.e. the point in the center of the triangle) would lead to the fair equilibrium of *Demand 5*. Despite this being exactly how the population is initialized with an Initial Moral Mean of 0.5 in the agent-based model, the results of the fully connected network are very different. It is not clear where this influence of the stochastic components comes from.

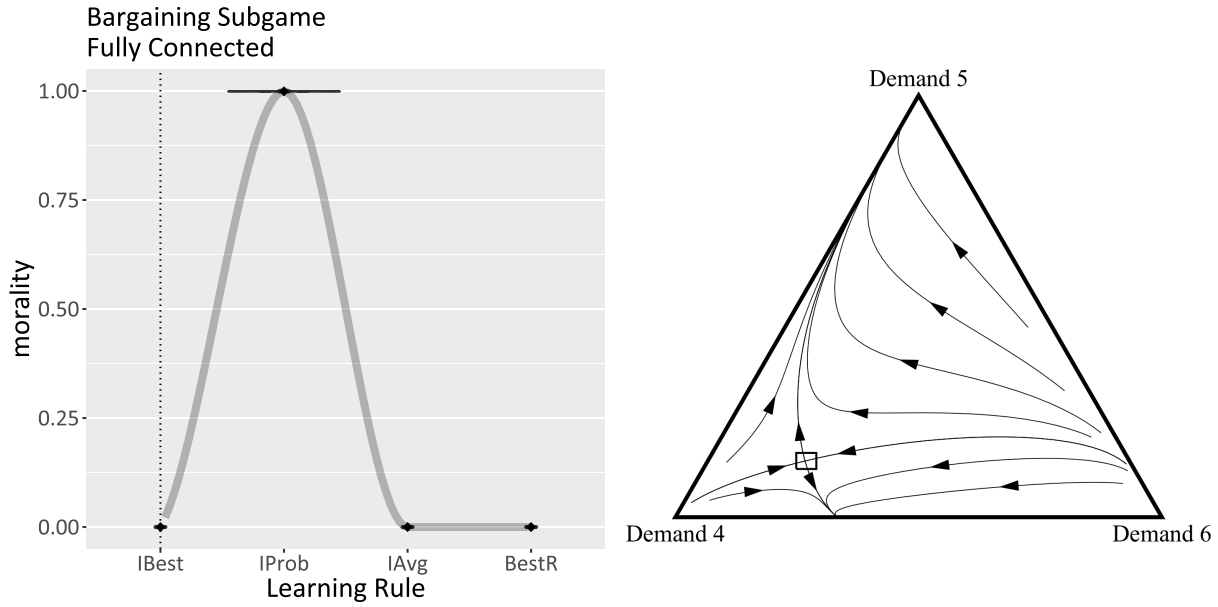

Figure 7.13: Results for the Bargaining Subgame on fully connected networks, and using the replicator dynamics (Alexander 2007, p. 156). Note that all results which are shown for the agent-based model use the nominal Initial Moral Mean of 0.5.

Ultimatum Subgame

For the Ultimatum Subgame, the simulations on the fully connected network always lead to minimum morality. Only a high Mutation Probability leads to a small increase in these results. This behaviour is visualized in figure 7.14. Alexander's experiments with the replicator dynamics and the Ultimatum Subgame show that morality emerges in about 15% of cases (Alexander 2007, p. 207). In contrast, for the simulations using the fully connected networks, the morality results are nowhere near 0.15. One explanation for the large difference could be the initializations. Alexander only mentions that "the initial conditions are selected at random" (ibid., p. 207). In the case of the agent-based model, the moral strategies *S5 (Easy Rider)* and *S7 (Fairman)* are each only chosen with probability 0.125 on average when using an Initial Moral Mean of 0.5. Increasing the mean to 0.75 presumably doesn't make much of a difference, given that it would only increase the probability to 0.167. The initialization chosen for the model is thus potentially counter-productive for games with many strategies, like the Ultimatum Subgame.

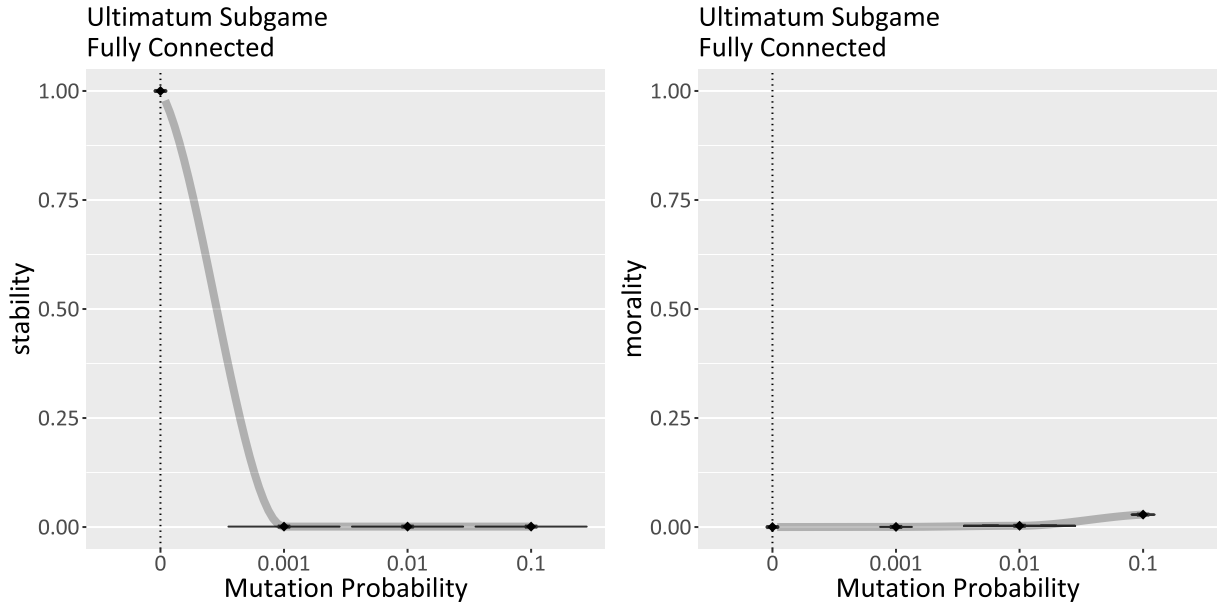

Figure 7.14: Results for the Ultimatum Subgame on fully connected networks, slightly influenced by the Mutation Probability.

7.3.3 Prisoner's Dilemma

In the Prisoner's Dilemma, the baseline result is minimum morality at maximum stability (shown in figure 7.15). No parameter is effective at significantly raising the morality measure above zero, except for Cooperation Incentive. The analogous Defection Incentive is not nearly as effective, as shown in figure 7.16. Interestingly enough, the results for the Cooperation Incentive parameter differ to a significant degree between the different network topologies, which cannot be said for most of the configurations of the model. Additionally, there is oftentimes a trade-off between morality and stability for these results. For example, the increased morality on the one-dimensional lattice comes with a lower stability measure (see figure 7.17). On the bounded-degree network, we see a proportionally weaker effect on both measures.

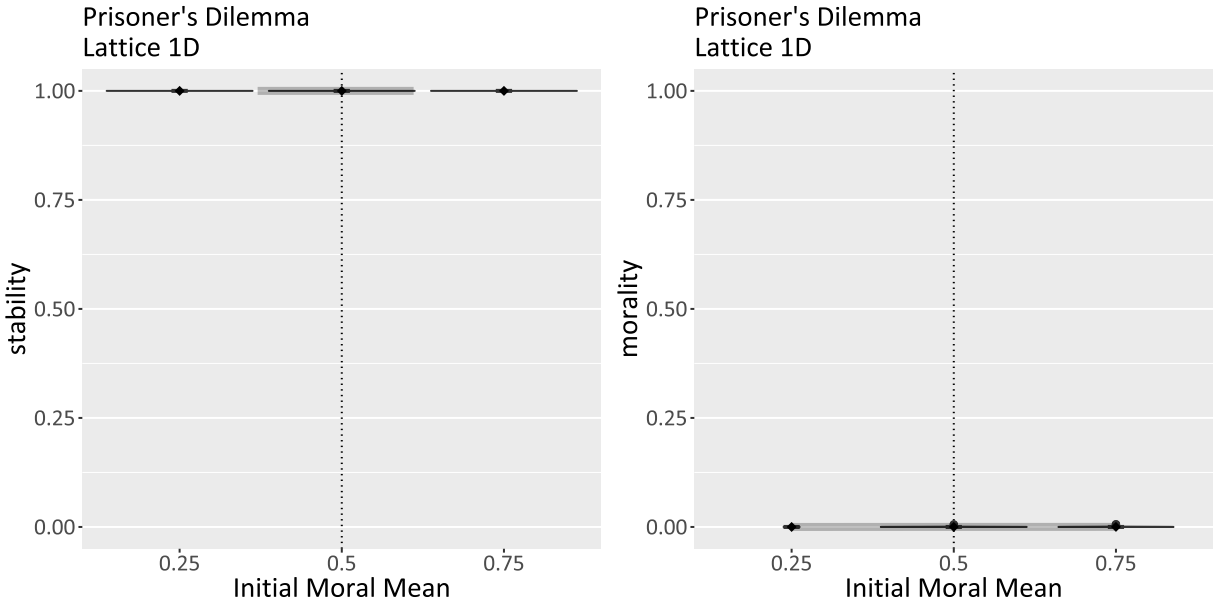

Figure 7.15: Baseline results for the Prisoner's Dilemma.

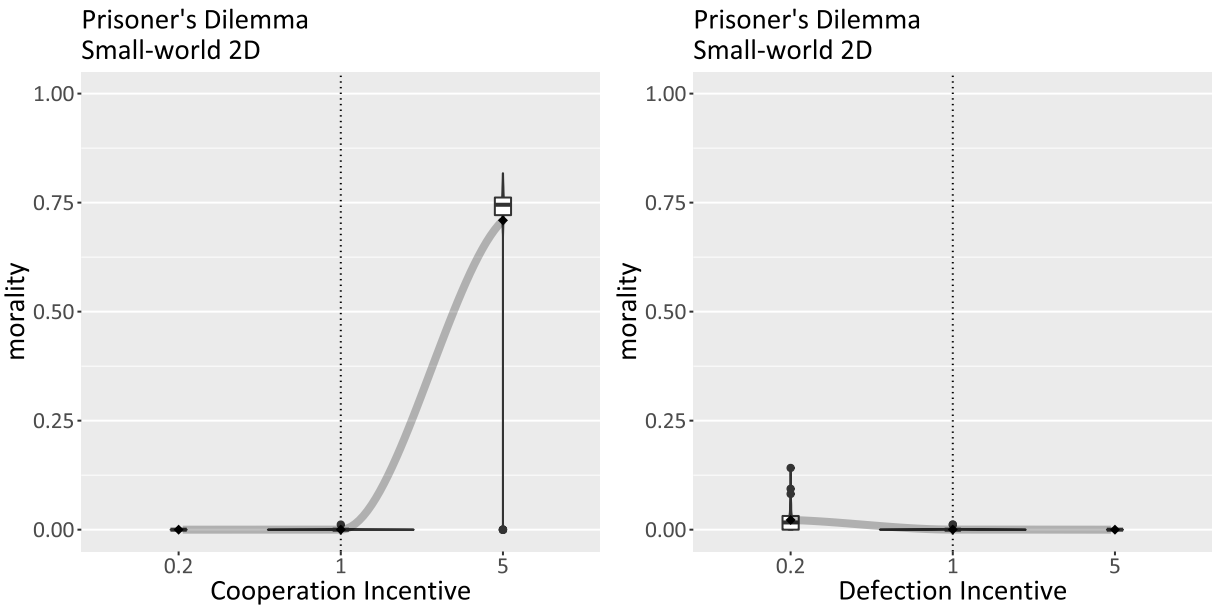

Figure 7.16: Comparison of Cooperation Incentive and Defection Incentive with regards to morality.

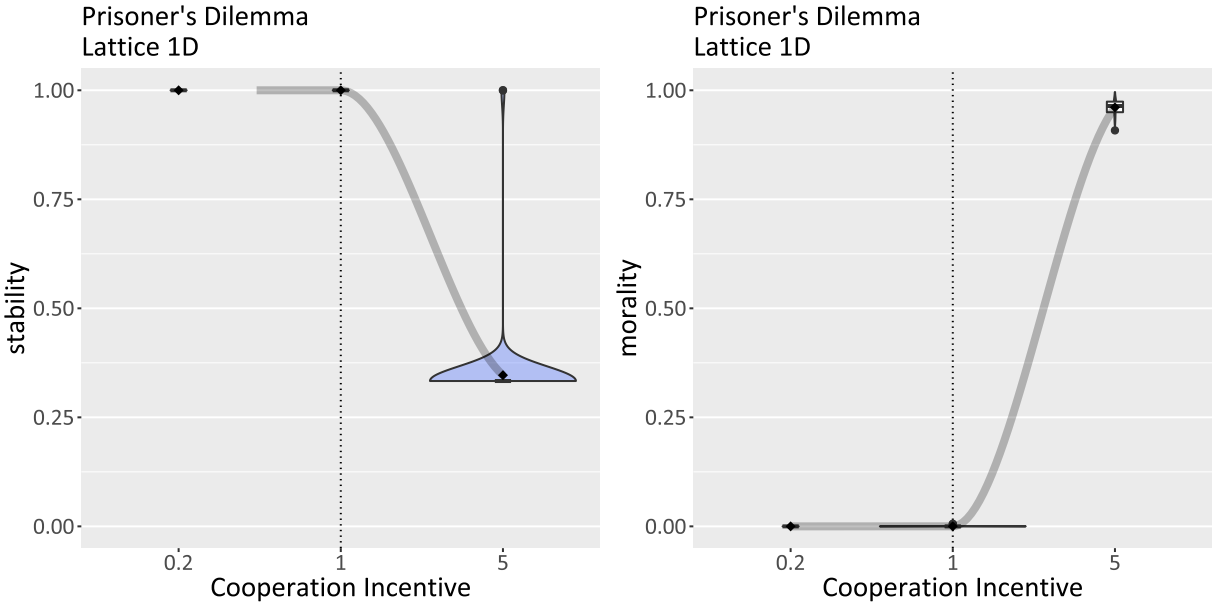

Figure 7.17: Influence of the Cooperation Incentive on stability and morality on the one-dimensional lattice.

7.3.4 Stag Hunt

For the Stag Hunt, the baseline result is maximum morality at maximum stability. While the Initial Moral Mean completely determines the result on the fully connected network, this doesn't happen anymore for other topologies. Low values for the parameter can still have a strong effect on some topologies, though, as shown in figure 7.18. While the learning rules *Imitate Best* and *Imitate Probability* generally lead to high morality results, *Imitate Average* sometimes leads to lower morality measures. *Best Response*, however, is extremely detrimental to morality in the Stag Hunt. As shown in figure 7.19, it consistently sets the mean of the morality result around 0.5, while producing very high stability results. Risk Dominance only has a noticeably effect for the one-dimensional small-world network. As mentioned before, the morality results for the Beta parameter of the one-dimensional small-world network show that small-world properties are comparatively bad for the emergence of morality in the Stag Hunt. Both behaviours are shown in figure 7.20.

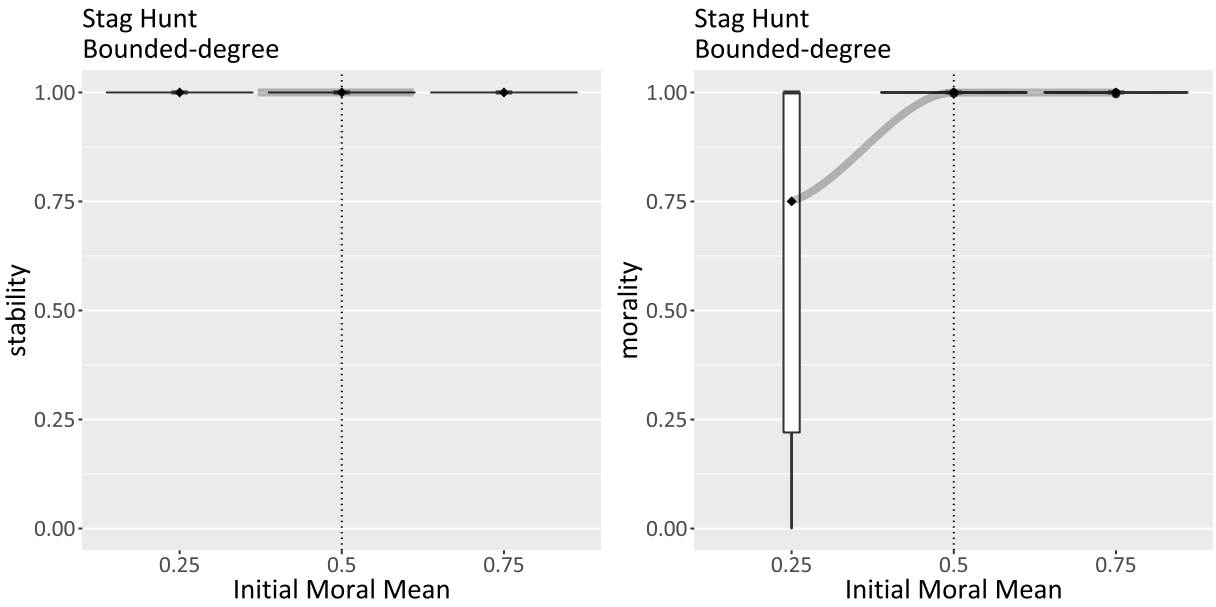

Figure 7.18: Baseline results for the Stag Hunt.

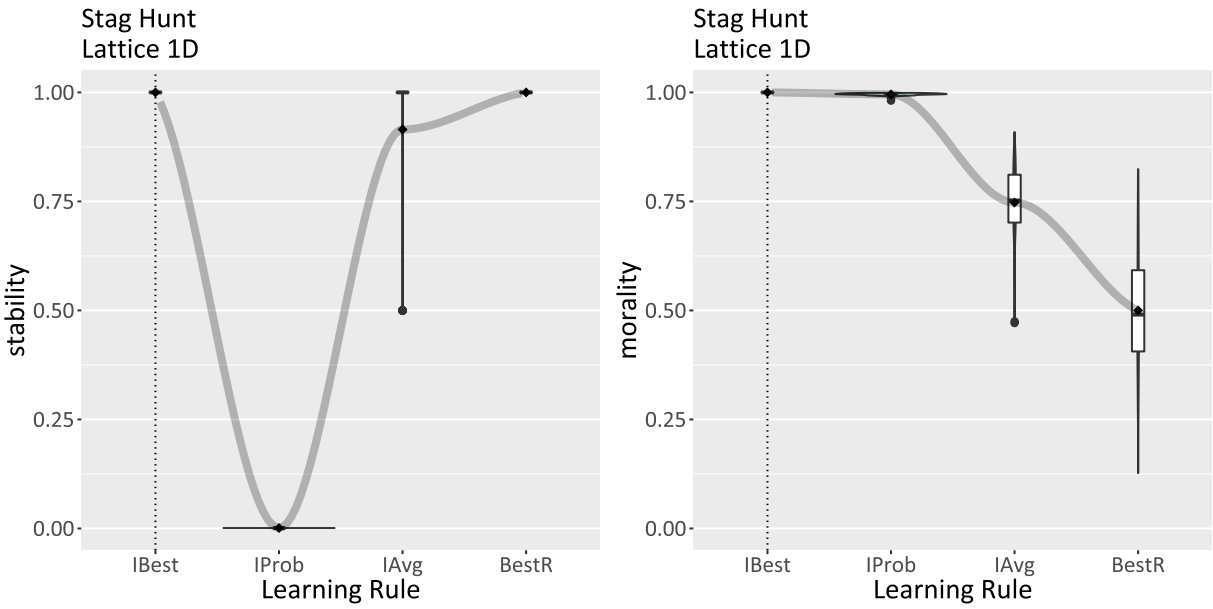

Figure 7.19: Influence of the Learning Rule on stability and morality in the Stag Hunt.

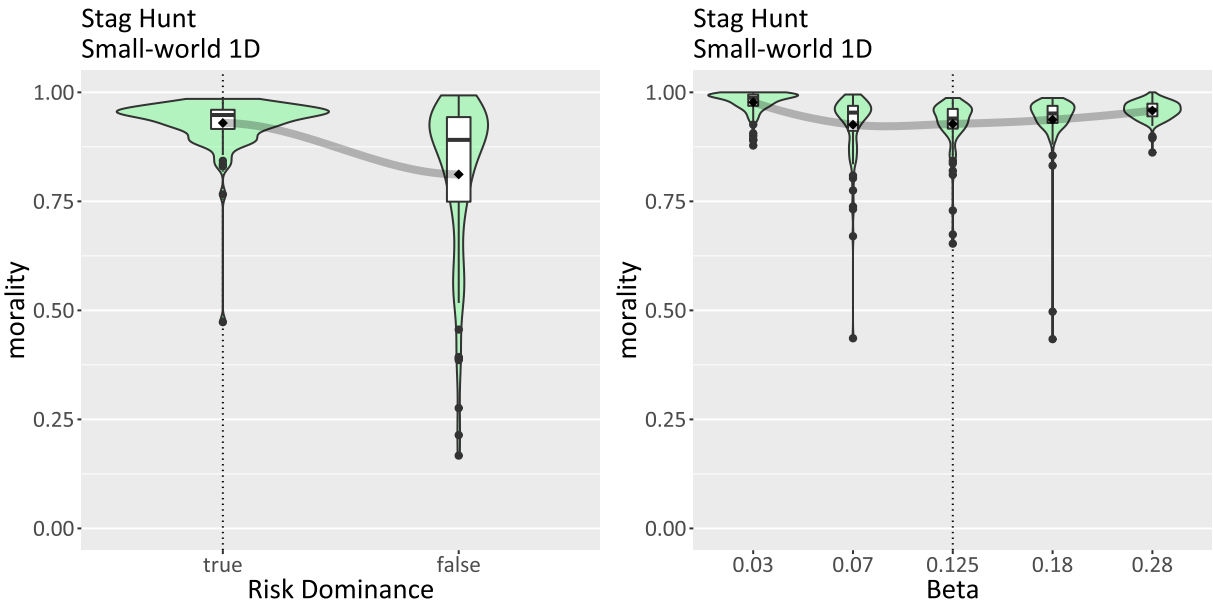

Figure 7.20: Influence of the Risk Dominance and Beta parameters on one-dimensional small-world networks in the Stag Hunt.

7.3.5 Bargaining Subgame

In the Bargaining Subgame, the baseline result is maximum morality at maximum stability (see figure 7.21). Similar to the Stag Hunt, the *Best Response* learning rule is very detrimental to morality. The corresponding stability measure is much lower, though, and hovers around 0.5. For this game, the small-world networks result in noticeably worse morality measures. They also have a much higher variance in their results compared to when being used in other games. Specifically the one-dimensional small-world networks exhibit unique behaviour (see figure 7.22). When using the latter topology, the results are strongly affected by the Population Size (stability drastically decreases with larger values), the learning rule (*Imitate Best* is worse for morality than *Imitate Probability* and *Imitate Average*) and Mutation Probability (no monotonic decrease in morality like usual). As mentioned before, the Beta parameter of the one-dimensional small-world network shows a negative impact of strong small-world properties on the morality measure (see figure 7.23). In the Bargaining Subgame, this effect is at its most extreme.

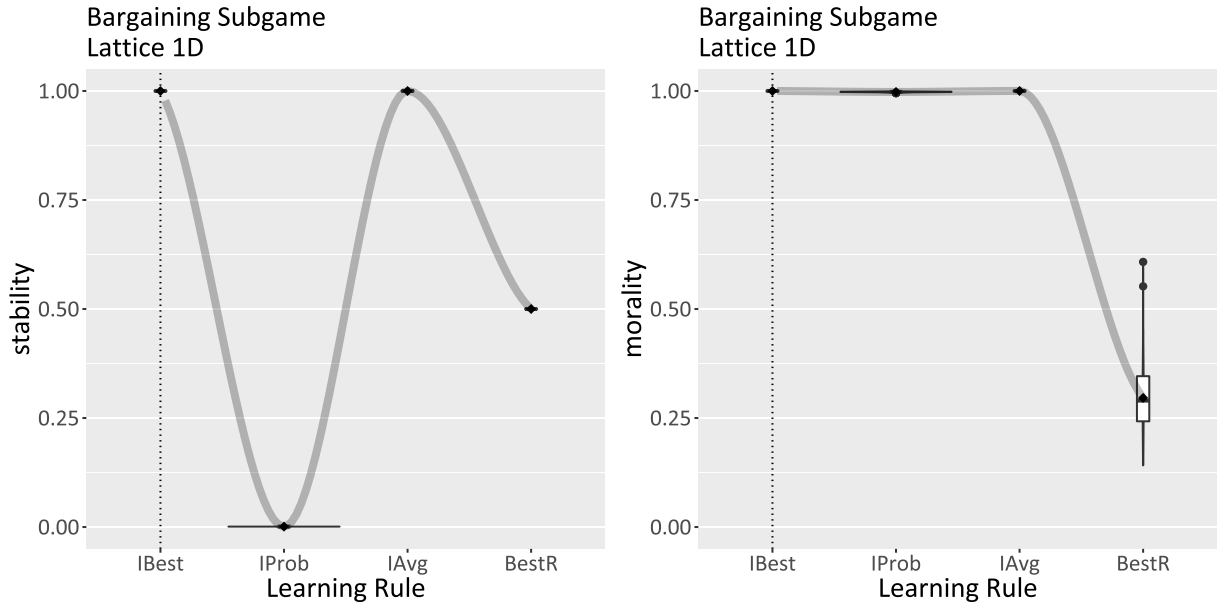

Figure 7.21: Baseline results for the Bargaining Subgame. Also negative impact of the *Best Response* learning rule.

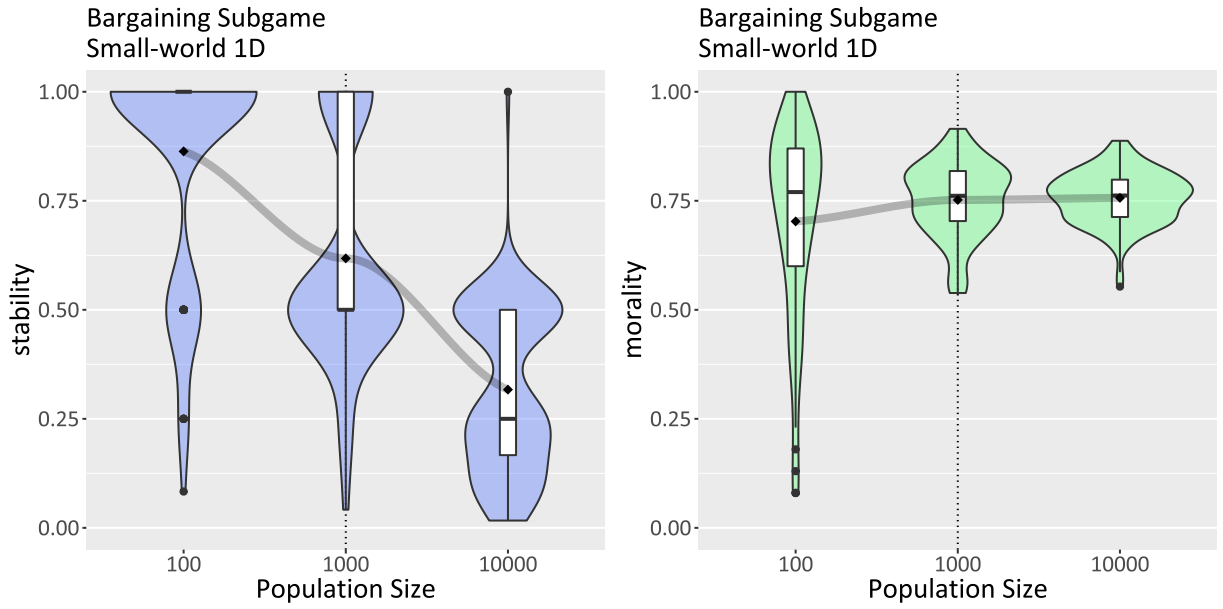

Figure 7.22: Influence of the Population Size on one-dimensional small-world networks in the Bargaining Subgame.

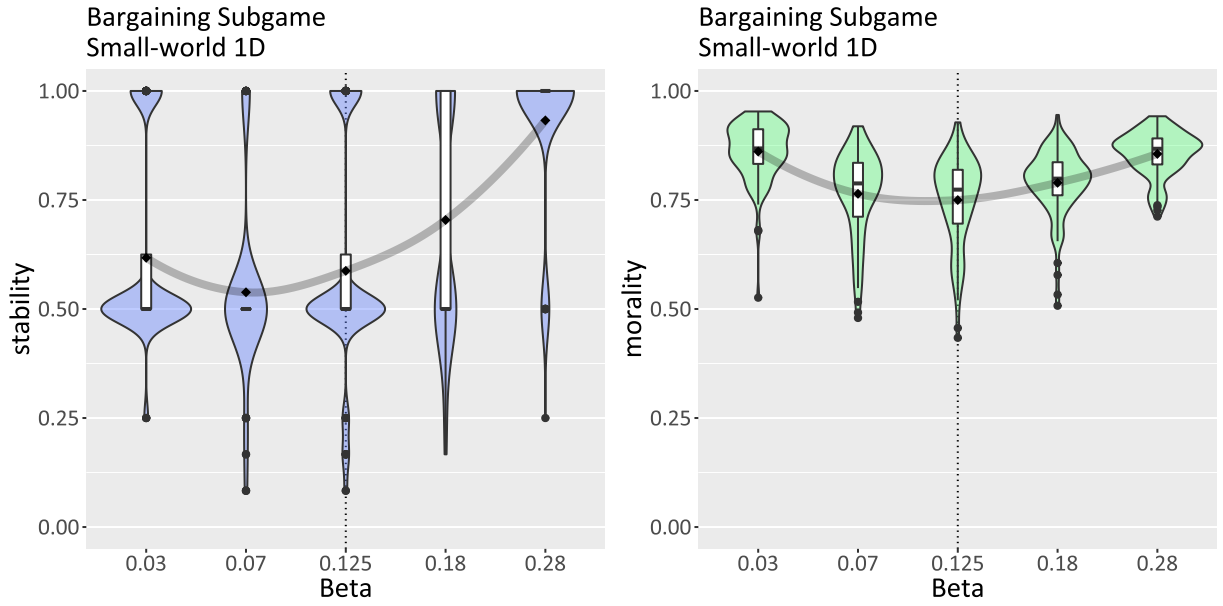

Figure 7.23: Influence of the Beta parameter on one-dimensional small-world networks in the Bargaining Subgame.

7.3.6 Ultimatum Subgame

For the Ultimatum Subgame, the baseline result is low morality and medium to high stability (see figure 7.24). Compared to the other games, the baseline result is not as accumulated at the extremes. There is more variability towards the center of the scale for both the morality and stability measures. In general, it is very hard to significantly increase the morality result. The maximum value among the tested configurations is around 0.3. While increasing the Mutation Probability typically moves the results away from the baseline for the other games, it has the opposite effect in the Ultimatum Subgame (see figure 7.24). As with both the Stag Hunt and the Bargaining Subgame, the *Best Response* learning rule has a negative impact on morality. The other learning rules do not exhibit this behaviour and share the same baseline. The one-dimensional small-world network has the same unique behaviour for the Population Size parameter as in the Bargaining Subgame. Increasing its value results in a sharp decline in the stability result (see figure 7.25). To a lesser extent, this is also true for the two-dimensional small-world network. The Ultimatum Subgame is the only game where the Neighborhood of the two-dimensional lattice makes any noticeable difference. The morality result is higher for smaller neighborhoods (N4 and M8, see figure 7.25). A similar effect can be observed for the one-dimensional lattice and both small-world topologies, but to a smaller extent and with a decrease in stability accompanying the increase in morality. The Beta parameter of the one-dimensional small-world network doesn't quite show the parable seen in the Stag Hunt and Bargaining Subgame. Instead, it has its maximum at the lower β and thus lower ω values, corresponding to more lattice-like topologies (see figure 7.26).

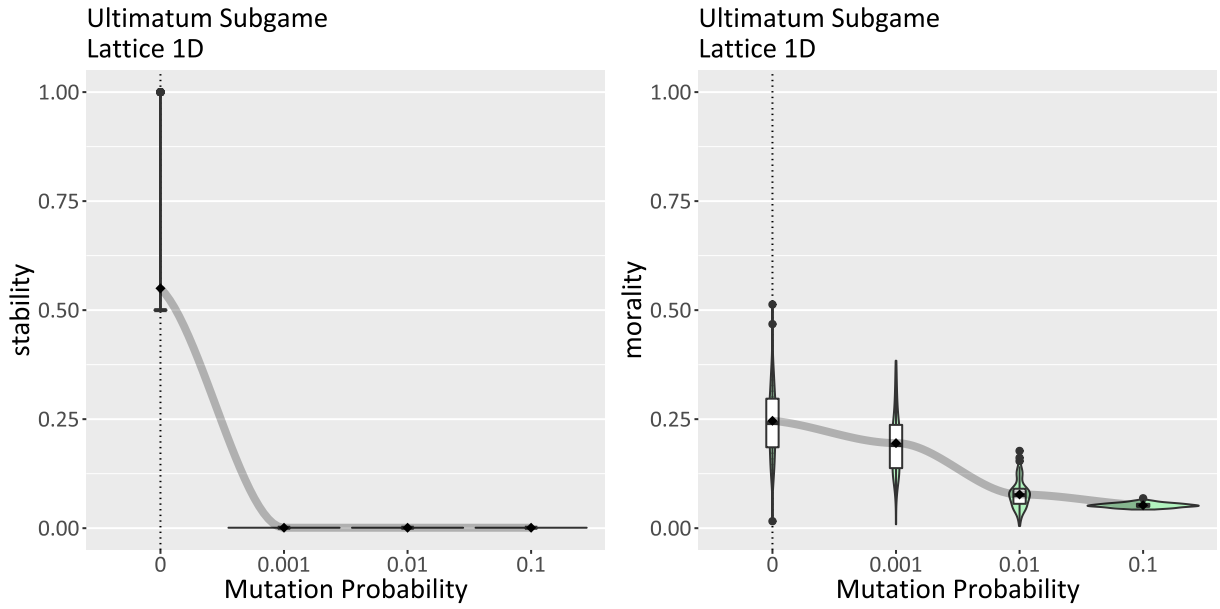

Figure 7.24: Baseline results for the Ultimatum Subgame. Also inverse influence of the Mutation Probability compared to other games.

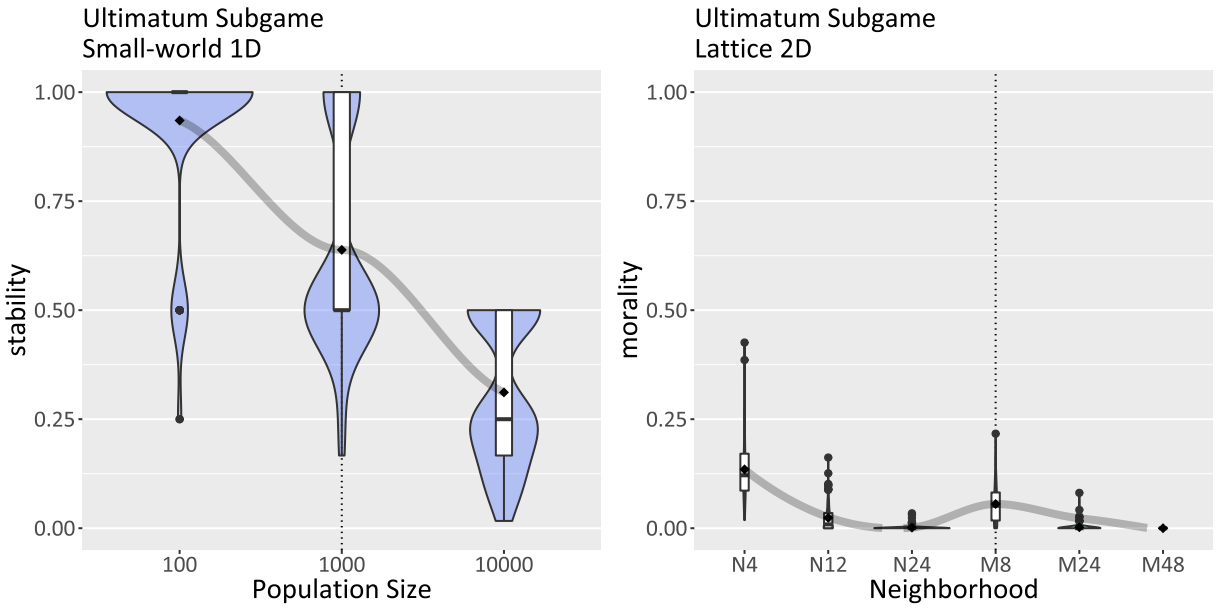

Figure 7.25: Influence of the Population Size on stability on the one-dimensional small-world network, and of the Neighborhood on morality on the two-dimensional lattice, in the Ultimatum Subgame.

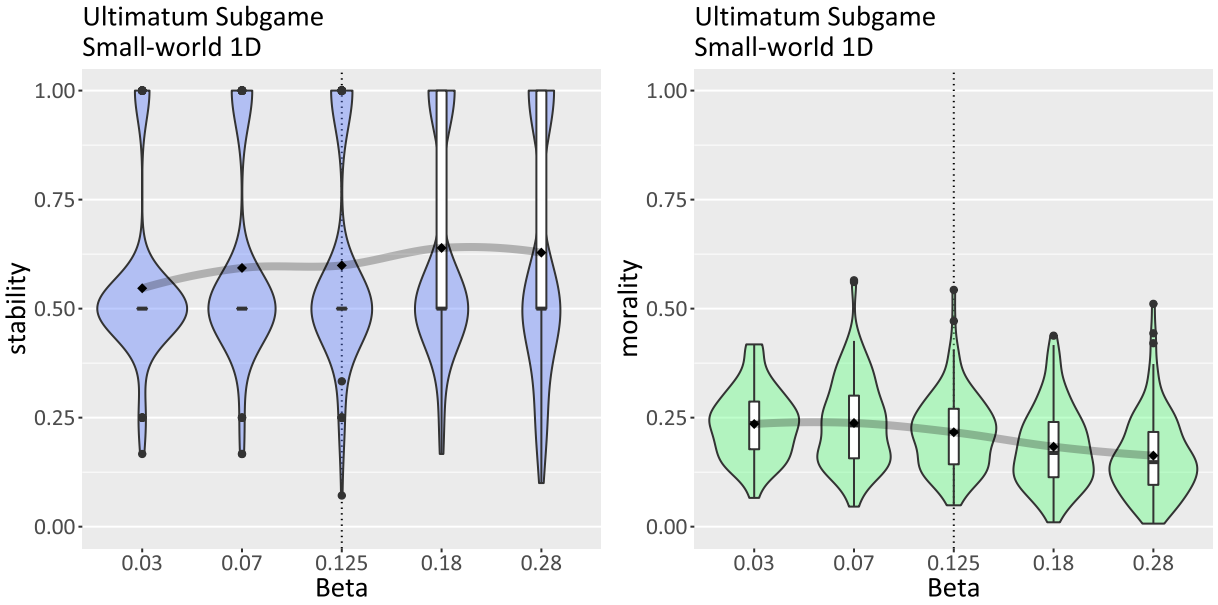

Figure 7.26: Influence of the Beta parameter on one-dimensional small-world networks on both the stability and morality measures in the Ultimatum Subgame.

Chapter 8

Discussion

This chapter discusses the model and the results of the sensitivity analysis with reference to *The Structural Evolution of Morality* (Alexander 2007), as well as future research on the topic of this paper in the area of agent-based modelling. Section 8.1 discusses the findings of the analysis in the context of Alexander's theory and compares the results to Alexander's experiments wherever possible. Section 8.2 presents some ways for future work to build upon the model and the analysis.

8.1 Findings

This section discusses the findings of the sensitivity analysis described in section 7.3 in the context of Alexander's theory, the Structural Evolution of Morality. Remember that as a result of the one-factor-at-a-time methodology used for the sensitivity analysis (described in section 7.1), the findings are restricted to the influence of one parameter at a time.

8.1.1 Practical Insights

There are a number of practical insights about the model. One hypothesis of the model was that the fully connected network topology could serve as a substitute for the replicator dynamics in agent-based models. This topology turns out to be a reasonably good approximation of the replicator dynamics. There are only two issues that came up with the results. First, in the Stag Hunt, the limited population size might have lead to a deviating tipping point between the two equilibria for *Stag* and *Hare* with regards to the initial distribution of strategies. While the replicator dynamics dictate that the population needs at least 50% *Stag* players to reach the moral equilibrium, the agent-based model results in a value somewhere between 50% and 75%. Second, in the Bargaining Subgame, the results of the fully connected network are the opposite of the replicator dynamics, whereby the former leads to minimum morality and the latter leads to maximum morality among the population. This deviation is removed when one activates mutation with a small Mutation Probability or uses the learning rule *Imitate Probability*. Given these results, the fully connected network seems to be a good approximation of the results

under structural constraints, except for the Bargaining Subgame without stochastic effects. This network topology can thus be viewed as a step between the replicator dynamics with infinitely large populations, and the other social network topologies with structural constraints.

Based on the results of the sensitivity analysis, some of the parameters turn out to have no practically relevant impact on the results *on their own*. This includes the Population Size and the Learning Distance in general, as well as the Neighborhood and Wrap Around of lattices. For the Learning Distance, we can imagine that it might have a noticeable impact when using games with more than 2–8 strategies, as the probability that every available strategy is part of an agent's neighborhood then rises with higher distances. Which neighborhood is used for a lattice, and whether it wraps around at the edges, presumably doesn't make a difference when the population is large enough. The takeaway here is that—as long as the combination of one of these parameters with other parameters does not have an important influence that is missed by analyzing one parameter at a time—these parameters can be removed from the model for increased simplicity and better performance.

8.1.2 Structural Constraints

The results of the sensitivity analysis indicate that the emergence of morality is still mostly determined by the shape of the interpersonal decision problem. Interestingly, which social network topology is used only has a significant impact when moral behaviour is not already the norm. For both the Stag Hunt and the Bargaining Subgame, the interactions on networks with structural constraints (i.e. not the fully connected network) generally lead to the desired outcome of *Stag* and *Demand 5* being the dominant strategies. The only way these results can be reliably and significantly overturned is by using the learning rule *Best Response*. This is the case for all network topologies. Thus, the specific topological constraints do not have a strong influence on the results of the Stag Hunt and the Bargaining Subgame. In the Prisoner's Dilemma and the Ultimatum Subgame, however, the analysis is more interesting. Here, the baseline result for structurally constrained populations is the undesired outcome of most agents playing *Defect* or *S1 (Gamesman)/S4 (Mad Dog)*. The influence of the specific network topology is significant in both games. For the Prisoner's Dilemma, the Cooperation Incentive parameter provides the only way to significantly increase cooperation among the population. This effect is strongest when using lattices or the fully random network. In the Ultimatum Subgame, the one-dimensional lattice and the one-dimensional small-world network provide the highest chance for moral behaviour to emerge.

8.1.3 Cooperation Incentive

Increasing the Cooperation Incentive, i.e. offering a much higher payoff for cooperation compared to defection while keeping the shape of the interactions in place, is the only way that cooperation in the Prisoner's Dilemma can emerge in the simulations. This generally conforms to Alexander's results: He explicitly mentions that specific payoff matrices are necessary for cooperation to emerge on lattices (Alexander 2007, pp. 72–75), small-world networks (ibid., p. 83) and bounded-degree networks (ibid., p. 89). In the case of bounded-degree networks, the

effect of the payoff matrix is lower compared to other topologies (ibid., p. 89), which coincides with the results of the sensitivity analysis: The morality result for Cooperation Incentive = 5.0 on social networks is indeed lowest for bounded-degree networks. Interestingly, the result for the fully random network is almost as good as that for the lattices.

8.1.4 Best-Response Reasoning

The *Best Response* learning rule causes interesting behaviour in both the Stag Hunt and the Bargaining Subgame, where it drastically worsens the morality results. In the case of the Stag Hunt, this behaviour has been analyzed in depth by Alexander. When using *Best Response*, the *Stag* equilibrium can only be reached when *Stag* is risk dominant (ibid., pp. 121–122). While this is theoretically guaranteed to happen in the limit as long as mutation is enabled (ibid., p. 126), it can take prohibitively long in practice (ibid., p. 128). Correlated mutation, i.e. using a higher Mutation Distance, can make the convergence to the *Stag* equilibrium much more likely (ibid., pp. 128–131). In the case of the sensitivity analysis, *Best Response* was only used in combination with the nominal parameters, which included neither the case where *Stag* is not risk dominant, nor the case where mutation is enabled. Interestingly, the *Best Response* learning rule also has a strong effect on the Bargaining Subgame. Here, moral behaviour emerges in even fewer cases than in the Stag Hunt. This configuration was only analyzed by Alexander on two-dimensional lattices (ibid., pp. 181–182). The sensitivity analysis uncovers that this result generalizes to all network topologies.

8.1.5 Small-World Networks

Small-world networks are often described as good approximations of true social network topologies, because their low average path lengths between two arbitrary vertices models the small degree of separation between two arbitrary people in the real world very well (Easley and Kleinberg 2010). Given that many people have strong moral intuitions about trust and fairness, one would expect that such network topologies are particularly conductive for the emergence of moral behaviour in the Stag Hunt and the Bargaining Subgame. The results of the sensitivity analysis, however, paint the exact opposite picture.

In the Bargaining Subgame, and to a lesser extent in the Stag Hunt, one-dimensional small-world networks exhibit a negative influence on the morality result. The extent of this negative impact is highest when the network has an ω measure that is close to zero. As mentioned in section 7.2, ω is a quantitative metric in the range $[-1, 1]$ which measures the small-world property of a graph. When ω is close to zero, this means a graph has maximal small-world properties. As ω decreases to -1, the graph looks more like a regular lattice, and as it increases to 1, it looks more like a random network. The center of the spectrum describes the small-world property, whereby a regular graph is augmented by few additional random edges, which lower the average path length between two arbitrary vertices. The more the one-dimensional small-world network actually exhibits strong small-world properties, the harder it is for trust and fairness to emerge, compared to more lattice-like and more random structures.

Reflecting upon these results, one could argue that the unfair division of resources is a seemingly common behaviour in our modern world. Thus, the strong negative influence of small-world networks confirms Alexander's theory, in that fairness did not properly evolve in human societies which are modelled by small-world networks. However, Alexander attempts to give an explanation for the strategic basis of the *moral intuitions* that many people share (Alexander 2007, pp. 273–275). Whether these intuitions are acted upon once they are in place is a different question. Still, one could argue that human societies modelled as small-world networks are disadvantageous to acting in a fair manner, regardless of whether one has a moral intuition about fairness or not. In any case, the connection between these results and the descriptive moral behaviour of humans today is worthwhile to investigate further.

8.1.6 Sparse Random Networks

When using bounded-degree networks, the Degree Interval $[2, 3]$ leads to special results. This particular value moves the morality result significantly away from the baseline in all games except the Prisoner's Dilemma. Both degree intervals that are wider (e.g. $[3, 8]$) as well as narrow intervals with higher degrees (e.g. $[8, 9]$), do not exhibit this behaviour. A similar, but weaker effect can be observed for the fully random networks when using an Edge Probability of 0.004. What special characteristics do these topologies have, that makes them have such a strong impact on the results?

The answer to this question is frankly unclear. Here, let us focus on the Ultimatum Subgame, because in this case the configuration increases the morality result. When manually experimenting, one can see that compared to other degree intervals, this configuration can more often lead to equilibria in which a sizeable portion of the population plays *S7 (Fairman)* while most of the other individuals play *S1 (Gamesman)*. This particular network topology seems to make it easier for clusters of moral agents to survive. The nature of the interactions can give some insight into why this might be the case. First, consider that when a *Gamesman* plays against a *Fairman* in both directions, both individuals get a payoff of 5. When a *Gamesman* is paired against another *Gamesman*, or when a *Fairman* plays with another *Fairman*, the two players get a payoff of 10 each in both cases. The two strategies are thus balanced out against each other very well. Second, the degree interval $[2, 3]$ leads to small neighborhoods. This increases the probability that clusters of *Fairman* players can survive, because they can reinforce each other in their strategic choice. They are not connected to many agents with other strategies that could lead them to change their strategies. Third, the lower connectivity of the network ensures that other strategies, which can be useful for the stability of *Fairman* clusters, are not immediately removed from the population at the beginning of the simulation. When, for example, an *Easy Rider* is paired with a *Fairman*, the former can oftentimes adopt the strategy of the latter, which in turn increases the size of the *Fairman* clusters. This provides a preliminary explanation for the observed behaviour in the Ultimatum Subgame. For further insights, one would have to investigate the interactions in more detail.

8.2 Future Work

There are many opportunities for future work to build upon the model and the results of the sensitivity analysis. This section presents some suggestions.

8.2.1 Model

The model offers a number of interesting potential alterations and extensions. In the Prisoner's Dilemma, it turned out that changing the numbers in the payoff matrix (while keeping the shape of the game in place) leads to strong differences in the results. The same concept could be applied to the other games as well. Especially for the Bargaining Subgame and the Ultimatum Subgame, it would be interesting to see whether different payoffs lead to different results. A simple way to test this would be to switch the extent of the payoffs between the two games, i.e. choosing the strategies *Demand 1*, *Demand 5* and *Demand 9* for the Bargaining Subgame, as well as strategies demanding and accepting portions of 4, 5 and 6 in the Ultimatum Subgame. It would also be interesting to see whether using the full Ultimatum Game makes any difference with regards to the morality and stability results. The Bargaining Subgame turned out to produce the same results as the full Bargaining Game in a number of experiments (see section 6.2.3). One should test this for the full Ultimatum Game as well. This would require larger populations, though, as the full Ultimatum Game contains many more strategies.

As we have seen in the cases when the results of the new model were compared to Alexander's experiments (see chapter 6 and section 7.3), different initializations of the population can have a strong impact on the dynamics of the interactions. Given this, an interesting avenue for future work is the incorporation of different kinds of initializations. One particularly promising approach would be to replicate the concept of the *evolutionarily stable state* from classic evolutionary game theory in an agent-based model. In this approach, the whole population is initialized with the same strategy. Then, a single agent (or a small cluster of agents) is assigned a different strategy. This way, it is possible to figure out whether the common strategy is evolutionarily stable (i.e. it cannot be taken over by a different strategy).

Focusing on Alexander's previous work, the model could be extended by adding structural dynamics through dynamic network topologies. The core idea is to model social networks as weighted complete graphs, such that the strength of a relationship between two agents can change over time by adjusting the corresponding edge weights. Alexander's results for dynamic networks are quite promising for the emergence of morality (ibid., e.g. 94-100). In order to keep the ability to compose arbitrary parameters of the model, the structural dynamics should be implemented for all the static network topologies offered so far. That is, when adopting the model for social network formation used by Alexander (Skyrms and Pemantle 2000), the network topology which the structural dynamics start from should be freely chosen. Based on this, one could add the additional features specific to dynamic networks which were used in *The Structural Evolution of Morality*: uncoupled frequencies for the learning and interaction processes (Alexander 2007, p. 52), as well as discounting of past interactions (ibid., e.g. 144-145).

8.2.2 Sensitivity Analysis

For the sensitivity analysis, there are many opportunities for further research as well. Some of the results are not easily explainable, and warrant a deeper analysis to understand them better. The Stag Hunt on a fully connected network produced a tipping point between the two equilibria (all agents playing *Stag* or all agents playing *Hare*) that is somewhat different from the ones delivered by the replicator dynamics (see section 7.3.2). The hypothesis is that the results would converge to the replicator dynamics when increasing the population size. This should be investigated. One-dimensional small-world networks have oftentimes lead to a high variance of the results, as well as producing the counter-intuitive detrimental impact on morality in the Stag Hunt and the Bargaining Subgame (see section 7.3.1). It would be interesting to know what causes the special behaviour of this topology, especially considering that the two-dimensional small-world networks do not behave this way at all. As mentioned in section 8.1.6, bounded-degree networks with a degree interval of $[2, 3]$ exhibit a very unique behaviour. While a preliminary attempt at explaining these results for the Ultimatum Subgame was offered, it is not clear why the effects can be generalized across the Stag Hunt and the Bargaining Subgame as well. This specific network topology warrants further analysis.

Apart from taking up specific findings, the analysis can also be improved by using a more sophisticated methodology for investigating the sensitivity of the different parameters to the result measures. As mentioned before, the one-factor-at-a-time method can only give insight into the effects of one parameter at a time, but not combinations of parameters. One could thus use a different method, like the Sobol' method (Sobol' 2001), to analyze the sensitivity of specific combinations of parameters. For example, in the Stag Hunt, Alexander's results show an important connection between the *Best Response* learning rule, the Risk Dominance of the *Stag* or *Hare* strategies, and the Mutation Probability and Mutation Distance (see section 8.1.4). Analyzing the combination of these parameters could provide further insight into this situation. Another interesting approach would be to see whether any combinations of parameters in the Prisoner's Dilemma provide a chance of fostering cooperation, without adjusting the payoff matrix through the Cooperation Incentive parameter.

Chapter 9

Conclusion

This paper built upon Alexander's results from *The Structural Evolution of Morality* (Alexander 2007). Alexander's analytically derived insights were complimented with a more systematic approach using numerical methods, in order to find out more about the exact conditions which need to be in place for moral behaviour to emerge. As described in chapter 2, related work in this area shows that models investigating the emergence of social norms in interpersonal decision problems oftentimes take different assumptions and use different parameters, which makes it hard to compare results among such models. Therefore, the approach taken was to create a new model that mimics Alexander's original model closely enough to facilitate comparison, while extending it in ways that allow for easy numerical analysis. The new model is focused on analyzing arbitrary combinations of games, networks, and other parameters. This makes it possible to use common methods of analysis from the agent-based modelling literature. After a short introduction into the theories that the model builds upon in chapter 3, the new model was described according to the ODD specification in chapter 4. Chapter 5 offered solutions to a number of challenges during the implementation of the model. The quantitative validation of the model against some of Alexander's experiments in chapter 6 showed that the two models produce results that are similar enough in trends to warrant a reasonable comparison. For the sensitivity analysis of the model reported in chapter 7, the one-factor-at-a-time (OFAT) methodology was chosen, which analyzes the influence of every single parameter on the results. While the approach is comparatively simple, it can already uncover interesting results and most importantly guide further analysis of the model.

The discussion of the analysis results in the context of the Structural Evolution of Morality in chapter 8 offered a number of interesting results. The emergence of moral behaviour is first and foremost determined by the shape of the interpersonal decision problem, leading to a baseline result for each game. Different network topologies can move the results away from the baseline to different degrees. The one-dimensional lattice is generally most successful at doing so. The specific payoff matrix of the game under investigation can play a big role in the emergence of moral behaviour, especially so in the Prisoner's Dilemma, where this is the only way to consistently create stable equilibria of cooperators. The more sophisticated *Best Response* learning rule, which requires individuals to reason counterfactually and stretches the assumption of bounded rationality, can be very detrimental to the evolution of morality. The

results of the simulations indicate that this is not only the case for the well-understood Stag Hunt, but also for the Bargaining Subgame. Small-world networks either don't impact the results in any significant way at all, or provide a negative influence on the emergence of morality. Especially in the Bargaining Subgame, the existence of strong small-world properties leads to significantly weaker moral behaviour among the population. Given that the topology is supposed to mimic real human societies, and that many people have strong moral intuitions about fairness, this is a surprising result in the context of the Structural Evolution of Morality. Finally, sparse random network topologies exhibit a very special behaviour which can lead to a strong positive influence on the emergence of morality in the Ultimatum Subgame, where it is typically hard to foster moral behaviour among the population. Both the model as well as the sensitivity analysis can be extended and built upon in numerous ways in future work. More research in this area can lead us to uncover the precise ways moral behaviour can evolve in populations of boundedly rational individuals engaging in interpersonal decision problems.

Bibliography

Alexander, J. McKenzie (2007). *The Structural Evolution of Morality*. Cambridge University Press. DOI: 10.1017/CBO9780511550997.

— (2019). "Evolutionary Game Theory". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Summer 2019. Metaphysics Research Lab, Stanford University. URL: https://plato.stanford.edu/archives/sum2019/entries/game-evolutionary/.

Axelrod, Robert (1984). *The Evolution of Cooperation*. Basic Books. ISBN: 0-465-02122-0. URL: https://www.basicbooks.com/titles/robert-axelrod/the-evolution-of-cooperation/9780465005642/.

— (1997). "Advancing the Art of Simulation in the Social Sciences". In: *Simulating Social Phenomena*. Ed. by Rosaria Conte, Rainer Hegselmann, and Pietro Terna. Springer Berlin Heidelberg, pp. 21–40. DOI: 10.1007/978-3-662-03366-1_2.

Axtell, Robert et al. (1996). "Aligning simulation models: A case study and results". In: *Computational & Mathematical Organization Theory* 1.2, pp. 123–141. DOI: 10.1007/BF01299065.

Balci, Osman (1994). "Validation, verification, and testing techniques throughout the life cycle of a simulation study". In: *Annals of Operations Research* 53.1, pp. 121–173. DOI: 10.1007/BF02136828.

Bandini, Stefania, Sara Manzoni, and Giuseppe Vizzari (2009). "Agent Based Modeling and Simulation: An Informatics Perspective". In: *Journal of Artificial Societies and Social Simulation* 12.4, p. 4. ISSN: 1460-7425. URL: http://jasss.soc.surrey.ac.uk/12/4/4.html.

Binmore, Kenneth G. (1994). *Game Theory and the Social Contract: Vol. 1, Playing Fair*. MIT Press. ISBN: 9780262023634. URL: https://mitpress.mit.edu/books/game-theory-and-social-contract-volume-1.

— (1998). *Game Theory and the Social Contract: Vol. 2, Just Playing*. MIT Press. ISBN: 9780262024440. URL: https://mitpress.mit.edu/books/game-theory-and-social-contract-volume-2.

Bollobás, Béla (1980). "A Probabilistic Proof of an Asymptotic Formula for the Number of Labelled Regular Graphs". In: *European Journal of Combinatorics* 1.4, pp. 311–316. DOI: 10.1016/S0195-6698(80)80030-8.

Bowles, S. and H. Gintis (2013). *A Cooperative Species: Human Reciprocity and Its Evolution*. Princeton University Press. ISBN: 978-0-691-15816-7. URL: https://press.princeton.edu/books/paperback/9780691158167/a-cooperative-species.

Britton, Tom, Maria Deijfen, and Anders Martin-Löf (2006). "Generating Simple Random Graphs with Prescribed Degree Distribution". In: *Journal of Statistical Physics* 124.6, pp. 1377–1397. DOI: 10.1007/s10955-006-9168-x.

Broeke, Guus ten, George van Voorn, and Arend Ligtenberg (2016). "Which Sensitivity Analysis Method Should I Use for My Agent-Based Model?" In: *Journal of Artificial Societies and Social Simulation* 19.1, p. 5. DOI: 10.18564/jasss.2857.

Davidsson, Paul et al. (2007). "Applications of Agent Based Simulation". In: *Multi-Agent-Based Simulation VII*. Ed. by Luis Antunes and Keiki Takadama. Springer Berlin Heidelberg, pp. 15–27. DOI: 10.1007/978-3-540-76539-4_2.

Deng, Lili, Wansheng Tang, and Jianxiong Zhang (2011). "The coevolutionary ultimatum game on different network topologies". In: *Physica A: Statistical Mechanics and its Applications* 390.23, pp. 4227–4235. DOI: 10.1016/j.physa.2011.06.076.

Easley, David and Jon Kleinberg (2010). "The Small-World Phenomenon". In: *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge University Press, pp. 537–566. DOI: 10.1017/CBO9780511761942.021.

Epstein, Joshua M. (2008). "Why Model?" In: *Journal of Artificial Societies and Social Simulation* 11.4, p. 12. ISSN: 1460-7425. URL: http://jasss.soc.surrey.ac.uk/11/4/12.html.

Erdős, Paul and Alfréd Rényi (1959). "On Random Graphs I". In: *Publicationes Mathematicae* 6, pp. 290–297. URL: https://snap.stanford.edu/class/cs224w-readings/erdos59random.pdf.

Gilbert, E. N. (1959). "Random Graphs". In: *Annals of Mathematical Statistics* 30.4, pp. 1141–1144. DOI: 10.1214/aoms/1177706098.

Grimm, Volker et al. (2006). "A standard protocol for describing individual-based and agent-based models". In: *Ecological modelling* 198, pp. 115–126. DOI: 10.1016/j.ecolmodel.2006.04.023.

Hamilton, W. D. (1967). "Extraordinary Sex Ratios". In: *Science* 156.3774, pp. 477–488. DOI: 10.1126/science.156.3774.477.

Ifti, Margarita, Timothy Killingback, and Michael Doebeli (2004). "Effects of neighbourhood size and connectivity on the spatial Continuous Prisoner's Dilemma". In: *Journal of Theoretical Biology* 231.1, pp. 97–106. DOI: 10.1016/j.jtbi.2004.06.003.

Klügl, Franziska (2008). "A Validation Methodology for Agent-Based Simulations". In: *Proceedings of the 2008 ACM Symposium on Applied Computing*. SAC '08. Association for Computing Machinery, pp. 39–43. DOI: 10.1145/1363686.1363696.

Lee, Ju-Sung et al. (2015). "The Complexities of Agent-Based Modeling Output Analysis". In: *Journal of Artificial Societies and Social Simulation* 18.4, p. 4. DOI: 10.18564/jasss.2897.

Luke, Sean, Gabriel Catalin Balan, et al. (n.d.). *MASON (Official Project Website)*. URL: https://cs.gmu.edu/~eclab/projects/mason/.

Luke, Sean, Claudio Cioffi, et al. (2005). "MASON: A Multiagent Simulation Environment." In: *Simulation* 81, pp. 517–527. DOI: 10.1177/0037549705058073.

Masuda, Naoki and Kazuyuki Aihara (2003). "Spatial prisoner's dilemma optimally played in small-world networks". In: *Physics Letters A* 313.1, pp. 55–61. DOI: 10.1016/S0375-9601(03)00693-5.

Matsumoto, Makoto (n.d.). *Mersenne Twister Project Website*. URL: http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html.

Nowak, M. and R. Highfield (2011). *SuperCooperators: Altruism, Evolution, and Why We Need Each Other to Succeed*. Free Press. ISBN: 978-1-4391-1017-1. URL: https://www.martinnowak.com/books/supercooperators.

O'Madadhain, Joshua, Danyel Fisher, and Tom Nelson (n.d.). *JUNG (Official Project Website)*. URL: https://jrtom.github.io/jung/.

Qawi, K. et al. (2011). "The Ubiquity of Small-World Networks". In: *Brain Connectivity* 1.5, pp. 367–375. DOI: 10.1089/brain.2011.0038.

Saltelli, A. et al. (2004). *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models*. Wiley. ISBN: 9780470870945. URL: https://www.wiley.com/en-us/Sensitivity+Analysis+in+Practice%3A+A+Guide+to+Assessing+Scientific+Models-p-9780470870938.

Sanchez, S. M. and T. W. Lucas (2002). "Exploring the world of agent-based simulations: simple models, complex analyses". In: *Proceedings of the Winter Simulation Conference*. Vol. 1, pp. 116–126. DOI: 10.1109/WSC.2002.1172875.

Seil, Stefan (2020). "Does the Structural Evolution of Morality Lead to Moral Relativism?" Bachelor's Thesis. LMU Munich, Faculty of Philosophy, Philosophy of Science and the Study of Religion. URL: https://github.com/srseil/Seom/blob/v1.0.0/literature/Seil2020.pdf.

Skyrms, Brian (1996). *Evolution of the Social Contract*. 1st ed. Cambridge University Press. DOI: 10.1017/CBO9780511806308.

Skyrms, Brian and Robin Pemantle (2000). "A dynamic model of social network formation". In: *Proceedings of the National Academy of Sciences* 97.16, pp. 9340–9346. DOI: 10.1073/pnas.97.16.9340.

Sobol', I. M. (2001). "Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates". In: *Mathematics and Computers in Simulation* 55.1, pp. 271–280. DOI: https://doi.org/10.1016/S0378-4754(00)00270-6.

Spiekermann, Kai P. (2009). "Sort out your neighbourhood". In: *Synthese* 168.2, pp. 273–294. DOI: 10.1007/s11229-008-9424-5.

Starnini, Michele et al. (May 2011). "Coordination and growth: the Stag Hunt game on evolutionary networks". In: *Journal of Statistical Mechanics: Theory and Experiment* 2011.05, P05008. DOI: 10.1088/1742-5468/2011/05/p05008.

Sun, Xin, Xishun Zhao, and Livio Robaldo (2017). "Ali Baba and the Thief, Convention Emergence in Games". In: *Journal of Artificial Societies and Social Simulation* 20.3, p. 6. DOI: 10.18564/jasss.3421.

Thiele, Jan C., Winfried Kurth, and Volker Grimm (2014). "Facilitating Parameter Estimation and Sensitivity Analysis of Agent-Based Models: A Cookbook Using NetLogo and 'R'". In: *Journal of Artificial Societies and Social Simulation* 17.3, p. 11. DOI: 10.18564/jasss.2503.

Vanderschraaf, Peter and J. McKenzie Alexander (2005). "Follow the Leader: Local Interactions with Influence Neighborhoods". In: *Philosophy of Science* 72.1, pp. 86–113. DOI: 10.1086/428077.

Watts, Duncan J. and Steven H. Strogatz (1998). "Collective dynamics of 'small-world' networks". In: *Nature* 393.6684, pp. 440–442. DOI: 10.1038/30918.

Weibull, Jörgen W. (1995). *Evolutionary Game Theory*. MIT Press. ISBN: 9780262231817. URL: https://mitpress.mit.edu/books/evolutionary-game-theory.

Wormald, Nicholas C. (1980). "Some problems in the enumeration of labelled graphs". In: *Bulletin of the Australian Mathematical Society* 21.1, pp. 159–160. DOI: 10.1017/S0004972700011436.

Xianyu, Bo (2010). "Social Preference, Incomplete Information, and the Evolution of Ultimatum Game in the Small World Networks: An Agent-Based Approach". In: *Journal of Artificial Societies and Social Simulation* 13.2, p. 7. DOI: 10.18564/jasss.1534.

Zhou, Lu et al. (2018). "Stag hunt and trust emergence in social networks". In: *Future Generation Computer Systems* 88, pp. 168–172. DOI: 10.1016/j.future.2018.05.053.

Declaration of Authorship

I hereby declare that I have composed the presented paper independently on my own and without any other resources than the ones indicated. All thoughts taken directly or indirectly from external sources are properly denoted as such.

Munich, February 26th 2021