PRIFYSGOL BANGOR UNIVERSITY
School of Computer Science
College of Physical & Applied Sciences

A Comparative Analysis of Evolutionary Game Theoretical Models Using Social Diversity And Single Strategies

Dorian-Benjamin Dressler

Submitted in partial satisfaction of the requirements for the
Degree of Bachelor of Science
in Computer Science

Supervisor Dr. Ik Soo Lim

May 2017

Acknowledgements

> *The aim of science is not to open the door to infinite wisdom, but to set a limit to infinite error.*
>
> — **Bertolt Brecht - The Life of Galileo**

I would like to thank Dr. Lim for his supervision and guidance, Dr ap-Cenydd for his tutorship and all staff and lecturers of the School of Computer Science for making these past three years a worthwhile and thought-provoking experience.

Statement of Originality

The work presented in this thesis/dissertation is entirely from the studies of the individual student, except where otherwise stated. Where derivations are presented and the origin of the work is either wholly or in part from other sources, then full reference is given to the original author. This work has not been presented previously for any degree, nor is it at present under consideration by any other degree awarding body.

Student:

Dorian-Benjamin Dressler

Statement of Availability

I hereby acknowledge the availability of any part of this thesis/dissertation for viewing, photocopying or incorporation into future studies, providing that full reference is given to the origins of any information contained herein. I further give permission for a copy of this work to be deposited with the Bangor University Institutional Digital Repository, and/or in any other repository authorised for use by Bangor University and where necessary have gained the required permissions for the use of third party material. I acknowledge that Bangor University may make the title and a summary of this thesis/dissertation freely available.

Student:

Dorian-Benjamin Dressler

Abstract

The emergence of cooperation is a phenomenon that has long puzzled scientists. Why do people cooperate when this entails foregoing a reward? Would purely selfish behaviour not be a better alternative? Network Science is a discipline which studies the dynamics of various networks, among these are Artificial Social Networks. In this context, Evolutionary Game Theory is used to model the dynamics that can cause the emergence of cooperation. In various models, actors are restricted to using a single strategy (cooperate or defect) for all of their interactions.

In recent years, Social Diversity was introduced as a mechanism to promote cooperation. This is based on the observation that humans use varied strategies in their encounters. Research suggests that this approach achieves higher bounds of cooperation than the thus-far standard approach.

This project seeks to scrutinise these claims, therefore the algorithms underlying these studies have been implemented. A comparative analysis of the obtained data has been carried out, for each of the models. These data sets provide an insight into dynamics that occur within social networks. This achieved a quantitative analysis under Ceteris Paribus conditions.

In addition, this project proposes a novel approach utilising group dynamics, which aims to model cooperation using social diversity in species that are less cognitively developed than humans.

The results indicate that social diversity is a viable mechanism that indeed increases cooperation, even if defecting yields a much higher reward. The novel approach was found to outperform these models.

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

Network Science is a multidisciplinary field which researches interactions in complex networks and spans a variety of domains such social networks, computer networks, communication networks and biological networks. Inspired by physics, mathematics, graph theory and biology, it is preoccupied with the representation and study of physical, biological and social phenomena. Its aim is to investigate the dynamics of these networks and to produce predictive models. The area this project focuses on is the emergence of the phenomenon of cooperation. Not only is its study fundamental to understanding human nature and that of other biological organisms, it can also be used to address problems in computing, such as resource sharing in P2P File-sharing Networks[10].

1.2 Aims

The aim of this project is to gain an insight into the discipline of network science, through research and replication of existing studies. It entails the development of analytic skills, which will be applied to offer a valid and contextual analysis of the interactions between rational actors that cause the emergence of cooperation.

1.3 Objectives

- To replicate evolutionary game theoretical models which see players use single and multiple strategies in their interactions.
- To subject these models to a series of tests that aim to measure their performance.
- To analyze the data and draw valid conclusions.
- To present a novel model expanding on the implemented models.

1.4 Hypothesis

The adoption of multiple strategies against different players in evolutionary game theoretical models achieves higher levels of cooperation than their single-strategy counterparts; and provide a more realistic context.

Chapter 2

Background

2.1 The conundrum of cooperation

Evolution describes a process in which organisms adapt to their environment in order to successfully compete for resources. This is reflected in the term “Survival of the fittest” [5] coined by Herbert Spencer and describes the process of natural selection proposed by Charles Darwin in ‘On the Origin of Species’[4]. How well an organism performs in its competition for resources serves as an indicator of its fitness. Based on this, one could assume that only the most selfish acting organisms manage to survive and thrive. However if fierce competition was the sole key to survival, we would soon find ourselves in a dilemmatic situation known as the Tragedy of Commons [7]. This arises when independent users of a shared resource act out of self-interest and behave contrary to the common good by collectively depleting it. Moreover, this is not congruent with what we observe in nature where examples of altruistic behaviour abound.

Altruism comes at a price, an altruist invariably foregoes a portion of its fitness and faces the risk of being exploited. This raises the question of how cooperation between independent actors emerges, in the absence of a central authority to enforce it.

2.2 Mechanisms enabling cooperation

In the absence of mechanisms to protect cooperators, defection will ultimately dominate a population. Martin Nowak, an eminent researcher in the field, summarises these key mechanisms in [12].

- Kin Selection: Describes the emergence of cooperation through genetic relatedness.

- Direct Reciprocity: Describes cooperation between unrelated individuals and even members of other species. Repeated encounters between players have been found to increase cooperation. This will be considered in more detail in the following sections.

- Indirect Reciprocity: While direct reciprocity relies on repeated encounters between players, indirect reciprocity takes into account the asymmetric nature of encounters. It encapsulates the idea of giving without reciprocation.

- Network Reciprocity: Players interact with each other in a spatial domain, such as a square lattice. This is considered in more detail in the following sections.

- Group Selection: Players can form groups, a group of cooperators may exhibit more success than a group of defectors. This specific approach has been the subject of controversy.

2.3 The Prisoner's Dilemma Game

The Prisoner's Dilemma Game (PD) is one of many models inspired by John von Neumann and Oscar Morgenstern's Game Theory [11]. It provides a mathematical framework to model the interaction between two independent,

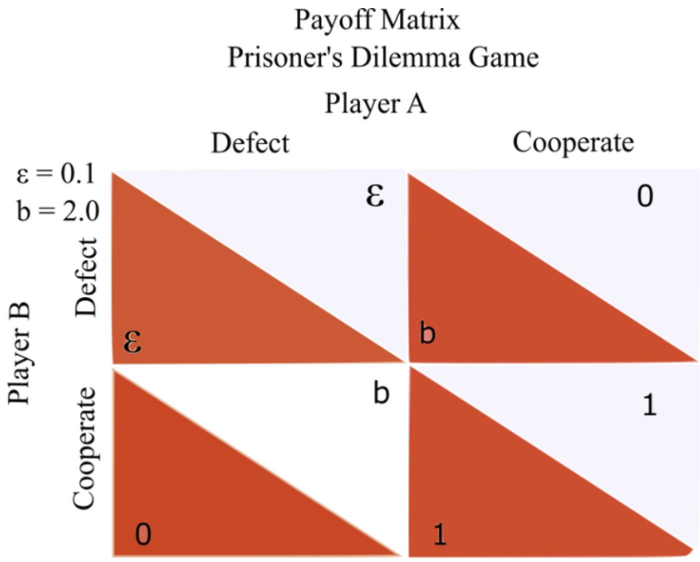

Figure 2.1: The payoff matrix of a 2 contestant Prisoner's Dilemma Game.

rational players, i.e. players acting in their own self-interest. Nowak identifies this model as the de-facto standard for games modelling the emergence of cooperation. In a theoretical single round game, two players – A and B – compete for a payoff. Each one can adopt a strategy to cooperate or defect (C, D) for this encounter. The combination of strategies determines the payoff each player receives. These consist of the payoff for mutual defection P, T for a (D, C) encounter, R denotes the payoff for reciprocal cooperation and S denotes the payoff in a (C, D) encounter. Note that the first term in the sequence is the strategy adopted by Player A, the second is Player B's. The value of these payoffs is strictly defined in the order $T > R > P > S$; this reflects the Darwinian model discussed in the previous sub-section. The game's payoff matrix is presented in Figure 2.1. The dilemma this game enshrines is that for both players the rational strategy is to defect, despite this furnishing both with a smaller payoff than cooperation would [1].

2.4 N-contestant iterated games

The stalemate dynamic in the PD can be resolved when the game is played iteratively, i.e. players encounter each other more than once. This is known as direct reciprocity. One prerequisite stated in the text is that players must not be cognizant of the number of rounds the game is played for, so to not adopt a strategy of defection at the next to last step. Another prerequisite stated is that players need to be able to identify another player and retain a memory of their last encounter.

In 1980, Axelrod issued a public invitation for the submission of iterative Prisoner's Dilemma games. A notable submission, and contest winner, was a game model named Tit For Tat, submitted by Anatol Rapoport[2], implemented with four lines of BASIC code. The two-player Tit for Tat follows the simple rule of "I do unto you what you do unto me". The game emerged as the winner of the contest, but suffered from a decisive flaw; encountering a defector would result in continued retaliation by a player through the adoption of defection (direct reciprocity). A later notable submission, Generous Tit for Tat, introduced the ability for players to forgive instead of remaining locked in continuous defection. Another successful game was Win-Stay/Lose-Shift which incorporates the simple idea of remaining with a successful strategy but altering it when it is no longer viable.

The n-iterated prisoner's dilemma game has since emerged as a standard approach for modelling social processes and is used in a variety of fields such as Conflict Resolution, Mathematical Psychology, Computer Science, Evolutionary Biology, and many more.

2.5 Monte Carlo Method

The Monte Carlo Method is used for statistical modelling based on random sampling [14]. Problems deterministic in nature benefit from the added uncertainty. The method utilises a Pseudo-Random Number Generator at its core [6] and is used to mimic operations in complex systems. In the context of evolutionary game theory, the method can be used in the player-selection process and ensures that opponent selection occurs non-linearly. This produces a more varied combination of interactions and results in an enhancement of ecological validity by providing broader variations of obtained data sets.

The method was developed to study systems in particle physics. Due the high similarity of dynamics, it is an appropriate tool to model cooperation in small, isolated populations equipped with a set of mechanism which simulate real-world behaviours on a small, observable and directly quantifiable scale. It finds application in economics, the network science, computer science, theoretical physics, conflict resolution and even psychology.

2.6 Artificial Social Network Topologies

2.6.1 Structured Populations

Spatial evolutionary games were introduced and investigated in [13], the authors modelled cooperation in cellular and molecular organisms. Players with no memory of past encounters are arranged on a N x N square lattice. Interaction is only possible with immediate neighbours and a cumulative payoff is accrued from these. As with previously discussed models, players maintain a strategy (C or D) in all their interactions and either remain with it or get replaced by a winning neighbour. The authors reported that the game resulted in chaotically changing spatial patterns they referred to as dynamic fractals and observed the formation of cooperative clusters. These clusters act as a defence mechanism against invasion by defectors and offer an explanation that reconciles the Darwinian model with the phenomenon of cooperation.

Inspired by this approach, the authors of the recently published paper [20] proposed a spatial game model utilising the Public Goods Game. This game model approximates the Prisoner's Dilemma game, but accounts for the increased number of players in a von Neumann neighbourhood [19]. Such a neighbourhood is composed of the central player and its four adjacent

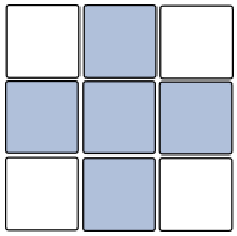

Figure 2.2: A von Neumann neighbourhood.

neighbours and is depicted in figure 2.2. Player selection occurs using the Monte Carlo method. A reputation system is introduced which allows a player to determine its fitness and that of its opponents, taking into account a variety of factors such as age. The authors found that this reputation mechanism combined with the spatial structure aids in successfully neutralising defection and results in an increase in cooperation. They posit that the model is inspired by the ‘credit system’ found in modern democratic societies. The model’s most poignant drawback: the use of a square spatial structure and the limited range of neighbours to interact with does not appropriately model the social structure and interactions among humans. The structured population is depicted in Figure 2.3

2.6.2 Well-mixed Populations

In contrast to structured populations is the well-mixed population. It enables random interactions between all players [8], its dynamics have been studied using various game models including the Snow Drift game and the Public Goods Game. In its basic, form the game does not incorporate mechanisms that protect co-operators from being exploited and simulations have shown that the game invariably results in a rapid elimination of cooperation. This population model is of significance because it can be used to illustrate the dynamics of cooperation and defection in large, interconnected populations and can be used to test various mechanisms that could promote the emergence of cooperation. A representation of a well-mixed population is shown in Figure 2.3.

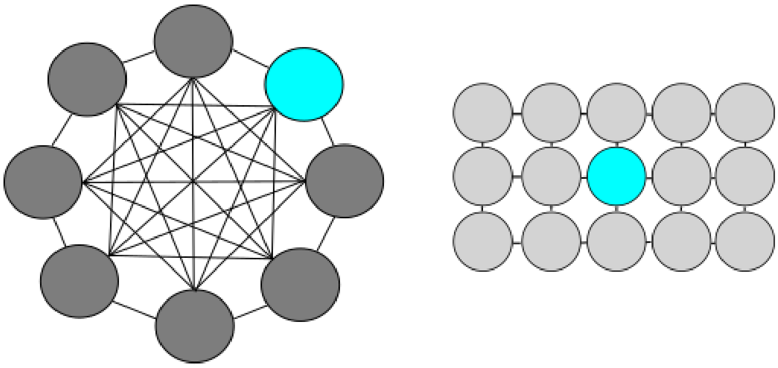

Figure 2.3: A side by side comparison of the well-mixed and the structured population of size N.

2.7 Games with social diversity

A relatively recent approach that aims to promote cooperation is social diversity [15]. It is based on the observation that individuals tend to use a diverse range of strategies against opponents rather than a single strategy for all. Wardil and da Silva [17], combine this approach with the spatial structure model discussed previously, noting that the latter was found to yield reproductive success for low defection tendency values. The research was carried out using 1D and 2D lattices, which restrict a player's scope of interaction to its nearest neighbours. Outcomes when using synchronous and asynchronous update rules were studied. These incorporate an imitation rule that takes effect when a randomly sampled focal player's cumulative fitness is less than that of a randomly chosen neighbour's. Three different replacement models were implemented and studied, one of them incorporates a mechanism replacing the strategy associated with the lowest contributor to the cumulative payoff. Findings reported are the emergence of cooperative clusters which retain bonds between cooperating players and neutralise attempts made by invading defectors. Along with high levels of evolutionary stable cooperation, even if a high tendency to defect is present; this tendency is referred to as b throughout. Additionally, they report lower, albeit stable, levels of cooperation if the model uses an asynchronous update strategy. Figure 2.4. shows players' using social diversity.

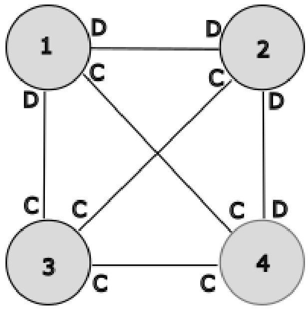

Figure 2.4: A simplified view of a population with players using social diversity.

Another study[18], conducted by the same team of researchers, tested the model's performance in a well-mixed population. The paper suggests that the model is strongly inspired by human interactions and aims to model the emergence of populations among them. Players are arranged on a linear lattice of length N, each player maintains N-1 strategies associated with all opponents. The rules remain the same as in the model previously discussed. A focal player is selected from the population, followed by a potential model. The focal player compares its cumulative payoff with the model's and adopts the model player's strategy if it has been outperformed by it. The strategy update occurs for the weakest contributor to the focal player's cumulative payoff. The scope of interactions now spans the entire population, excluding the player. Reported results are that high levels of cooperation are achieved for the synchronous update method, the asynchronous update manages to maintain existing levels of cooperation, after neutralising defection.

The model serves as an interesting metaphor, but exhibits a flaw by assuming an unrealistic level of connectedness. While humans are indeed a highly connected species, they do not share a connection with every single member in a population. Moreover, maintaining a strategy for every single interaction is a memory intensive task and would therefore come at a very high cognitive cost.

2.8 Players using social diversity in Strategic Groups

This model is inspired by the well-mixed population model, proposed by Wadil et al. This novel approach adapts this model in a way that excludes the need for high cognitive abilities, so to model the emergence of cooperation in species with lower cognitive abilities. To achieve this, the model will incorporate a group selection mechanism, i.e. a player will not maintain a strategy for each opponent. Instead players segregate the population into groups and maintain a strategy associated with each. The question of this research is whether such a model produces similar if not higher levels of cooperation under the same conditions. Figure 2.5 illustrates this model.

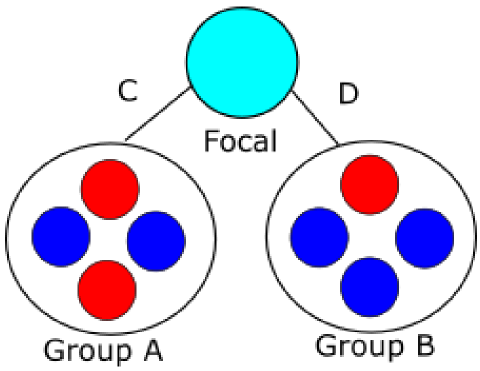

Figure 2.5: The model with 2 opposing groups. Blue nodes are cooperators, red nodes defectors.

Chapter 3

Implementation

3.1 Technologies used

- Java: Java is an Object Oriented high-level general-purpose Programming Language, released by Sun Microsystems in 1995. In the scope of this project, it was used to implement the algorithms underlying the game models and produce measurable outputs.

- Maven: Apache Maven is a software project management and comprehension tool. Based on the concept of a Project Object Model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

- Github: Github is a web-based version control tool. It was used to keep track of different versions and to potentially make the implementation available to the general public.

- MATLAB: MATLAB by MathWorks is a programming language used for matrix multiplication, computer graphics and data visualisation. The data sets produced by the game models is visualised using this language.

3.2 Implemented Game Models

3.2.1 Model A: PD in Well Mixed Population

The population in this game model is represented by a ring of size N. Each node maintains a single strategy. The algorithm is described below.

```
set population to 50/50 C/D;
set game rounds size to 100,000;
set population size to N;
for set i to 0, i < game rounds; increment i do
    set focal node's cumulative payoff to 0;
    set model node's cumulative payoff to 0;
    select random focal node from population;
    for each node in population do
        if node is not player then
            play prisoner's dilemma game with opponent;
            add payoff to focal's cumulative payoff;
        else
            skip node;
        end
    end
    for each node in population do
        select random unique model node from population;
        if node is not player then
            play prisoner's dilemma game with opponent;
            add payoff to model's cumulative payoff;
        else
            skip node;
        end
    end
    if model cumulative payoff > focal cumulative payoff then
        update strategy;
        set focal strategy to model strategy;
    else
        do nothing;
    end
end
```

Algorithm 1: Base algorithm for single strategy game

3.2.2 Model B: Single Strategy PD in 1D and 2D Structured Population

The population consists of nodes arranged on a square lattice of linear Size N. Nodes maintain a single strategy for connections it shares. A node situated in the population interacts with players that are directly adjacent to it. For example, player A is situated at x,y and can only interact with players in its 4-connected von Neumann neighbourhood. Each round of the iterated game begins with the selection process. A focal player A is randomly selected from the population. This is followed by the selection of a model player B from the focal player's immediate neighbourhood, within the pre-specified constraints. Player A will then play the Prisoner's Dilemma game with all nodes it shares a link with, player B will do this as well. The resulting individual pairwise payoffs are summed and the cumulative payoff is taken to be the node's fitness. A strategy update stage is effected when the model's fitness surpasses the focal player's. This strategy update sees the focal player adopting the strategy used by the model in its future interactions. If the cumulative payoff of B is <= A, focal Player A will remain with its original strategy. The algorithm resembles the algorithm used by Model A.

3.2.3 Model C: Different Strategy PD in 1D and 2D Structured Population

The population consists of nodes arranged along a square lattice of linear size N. Individual nodes can only interact with other nodes within their immediate neighbourhood. An immediate neighbourhood is defined in the x,y-space as a 4-connected von Neumann neighbourhood. Special cases that consider nodes located at the boundaries of the population were implemented. An individual node maintains a strategy associated with each of its neighbours. During the game's selection stage a random focal node A is selected from the population. A random model opponent B is selected from the node's immediate neighbourhood. A and B play the Prisoner's Dilemma game with each opponent in each of their respective neighbourhood. During such an interaction, a node realises a pairwise payoff. This payoff is subject to the strategy the player uses against the opponent, and conversely the strategy the opponent uses against the node. A cumulative payoff is realised from the sum of all these interactions, with a node maintaining a list of the individual contributions the other nodes have made. The resulting cumulative payoffs of Player A and B are compared, and if cumulative payoff A < cumulative payoff B, a strategy update is carried out.

If an update occurs, Player A copies the strategy B used against it, in their interaction and will use it to replace its strategy associated with the node contributing the lowest pairwise payoff. If more than one interaction yields the lowest contribution, one of these nodes will randomly be selected for update. The update condition statement differs to the previous models and is shown under Algorithm 2.

if *update condition satisfied* **then**
    identify lowest contributor from adjacent nodes;
    update link to lowest contributor with strategy used by model;
else
    return to top;
end

Algorithm 2: Update statement for structured multi-strategy model

3.2.4 Model D: PD with N-1 different strategies in Well-mixed Population

The population is structured in a Ring of size N and sees all nodes interconnected. Each node maintains a strategy vector of size N-1 which contains the strategies it uses when interacting with its opponents. This strategy vector satisfies the condition that defection and cooperation strategies are present in a 50 percent configuration. During the selection and interaction stages a random focal player A is selected from the population.

Player A realises a cumulative payoff from all N-1 interactions, by playing the Prisoner's Dilemma game considering the strategy it associates with each individual player and the strategy each player uses against it. A random, unique opponent B is selected as the model. The model player realises a cumulative payoff, signifying its own fitness. Both payoffs are compared to see if the update condition has been satisfied. An update occurs if the cumulative payoff B > cumulative payoff A. In this case, the focal player will adopt the strategy the model used against it in their previous interaction and apply it against the weakest contributor to its cumulative payoff, in a future round. For example, player B uses D in a (C,D) encounter – where C denotes the strategy Player A uses against B and D being the strategy Player B uses against A. Player A then identifies a node which provided the lowest contribution to the cumulative payoff. To do this, it traverses the individual payoffs and takes note of the nodes which provided the least amount. If more than such a node is present, the model requires the random selection of one of these. It then replaces the strategy it previously associated with the link to the weak provider with the model's strategy. The model player remains with its own strategy if the update condition has not been satisfied.

3.2.5 Model E: PD with Strategic Groups in Well Mixed Population

(a) with two groups:

This model is inspired by the previously discussed model. The population model sees the nodes arranged on a Ring of size N. In a simplified setup, each node splits the populations into two groups of equal size. A node n_i is situated at the centre of the group, with group Group A containing a set of nodes from $n_i(N/2)$ to $n_i - 1$ and group Group B containing all nodes from $n_i + 1$ to n. In this scenario, a node will only maintain two strategies, namely the strategies associated with group A and group B. The population is created with an initial cooperation ratio of 50 percent, meaning that a player is equally probably to encounter a defector a co-operator. During the selection stage, a random focal player A is selected from the population. Player A plays the Prisoner's Dilemma game with each other node in the population. To do this it verifies which group the opponent player belongs to and applies the appropriate strategy. The selected opponent in turn will supply the strategy it associates with Player A. The resulting pairwise payoffs are stored and a cumulative payoff is realised when Player A finishes its interaction with all N-1 opponents. In addition to this, a focal player is aware of the individual cumulative payoffs it received from each group. A model player B is selected from the population which calculates a total cumulative payoff of its own. The cumulative payoffs achieved by A and B are consequently compared to see if they satisfy the update-condition. This update condition is specified as a cumulative payoff A < cumulative payoff B, where each variable identifies the player's fitness. If the fitness of Player B exceeds that of Player A, Player A will adopt the strategy used against it by B and apply it in future interactions against the group which provided the lowest cumulative payoff. In a scenario where both groups contribute the same amount, the model will select one of them at random for the strategy replacement. If the focal player's fitness is equal to or exceeds that of the model, the focal player will remain with its original strategy. Pseudocode for this Algorithm is shown under Algorithm 3.

(b) with N/2, N/4, N/8 groups A well mixed population is set out on a ring of size N. Each node maintains a strategy (cooperate or defect) associated with a group. Each node can identify which group another player belongs to and selects the appropriate strategy for its interaction with this group. Strategies used for groups are initialized at a 50/50 defection/cooperation ratio. The setup of these strategic groups depends on the number of groups each node maintains. A simplified example sees a node maintaining N/2 groups and sees the population split into N/2 equal sized groups. These groups include all opponents, except the player itself. The size of a group is defined by the size of the population/number of desired groups. During selection and interaction stage, a focal player A is randomly selected from the population. It plays the Prisoner's Dilemma game with all other opponents (excluding itself). When it encounters an opponent, it checks which strategic group this opponent is in and selects the appropriate strategy. The opponent provides the strategy it uses against the group Player A is in. The player keeps track of the payoffs it receives from each group and stores these in a different dataset. A model player B is selected, and it plays a round of the game with each of its opponents to realise a cumulative payoff. These cumulative payoffs are tested to see if the update condition cumulative Payoff A < cumulative Payoff B has been satisfied. If this is the case, Player A will copy the strategy Player B used against it in their last encounter, traverse its group-wise payoffs to find the lowest contributors and apply the model's strategy to the lowest contributor for their next encounter. If more than a single group-wise payoff is found to contribute only a small amount, the model requires that a random one of these is being selected. If the update condition has not been satisfied, Player A will not alter its strategy. Larger group sizes resulting in less groups can be realised by applying this model.

3.3 Design Considerations

The implementation is used to simulate iterated game theoretical models, simulating cooperation in populations and to evaluate their performance. The program was not intended for end-users without knowledge of the Java Programming Language. Users familiar with the language should find it easy to implement their own game models using the provided Utility Package. The implemented game models are contained in the GameModels package. This package also contains the class 'Node'. In the context of this simulation a node is akin to a player in the game. Nodes are stored in a data-structure inside their respective game model class. The type of data structure represents the network's topology. Notifications, and Experiment Constants are provided in the Utility.pkg

The way these classes interact is shown in Figure 3.1.

```
set population size to N;
set game rounds to 100,000;
for each node in the population do
    segment opponents into two groups left and right of node;
    randomly assign C or D strategy to each group;
end
for i < game rounds; i++ do
    set focal node's total cumulative payoff to 0;
    set model node's total cumulative payoff to 0;
    select random focal node;
    select random model that is not focal;
    Repeat following step for focal and model
    for model and focal :  traverse each node in population do
        if node is not player then
            Obtain group opponent is in;
            Set group strategy to strategy associated with group;
            Get opponent's strategy;
            Play prisoner's Dilemma game with opponent;
            Add payoff to relevant accumulated group payoff;
            Add payoff to total accumulated payoff;
        else
            skip node;
        end
    end
    if total accumulated payoff focal < total accumulated payoff model then
        set update strategy to strategy used by model against Focal;
        determine group providing lowest contribution to model total
          accumulated payoff;
        replace strategy used against group with new strategy;
    else
        do nothing;
    end
end
```

Algorithm 3: Algorithm for Group Strategy Game with two groups (Model E (a))

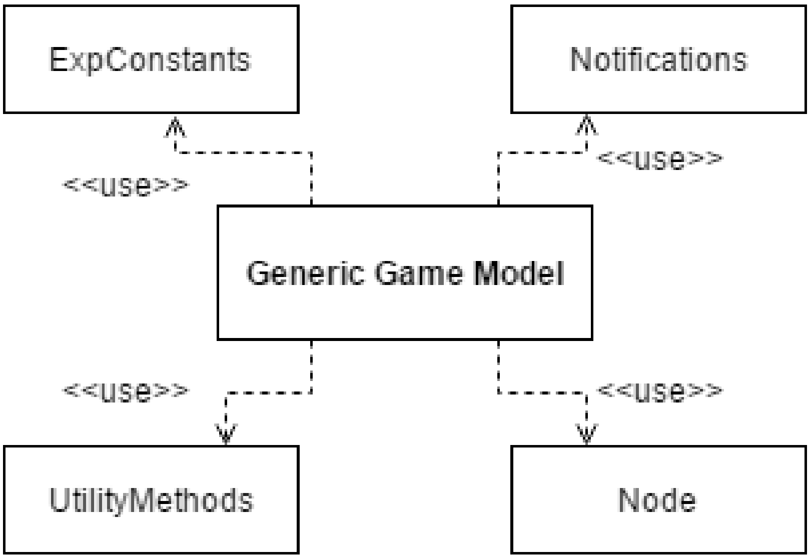

Figure 3.1: Interactions of Classes

3.3.1 The Utility Package

Three classes comprise the Utility Package: ExpConstants.java, Notifications.java and UtilityMethods.java. These were designed to provide static methods and static variables that can be imported into game models and provide standard methods for a host of tasks, used repeatedly across the implemented simulations. Notifications contains standard notifications such as Error Messages and helps to provide a standardised interface. ExpConstants contains constants to be used in the game model classes. This enables users to adjust variables used by game models, without having to make any edits to the base classes. Figure 3.2 shows the class diagrams for this package.

Of note is setup2DPopulation(), it can take varied parameters which include the linear size of the desired array. It will generate a 2 dimensional array populated with nodes. For this it checks whether the population size is a power of two. If not, it uses prime factorisation to generate an appropriate array. Should this not be possible, a brute force method is called after five passes.

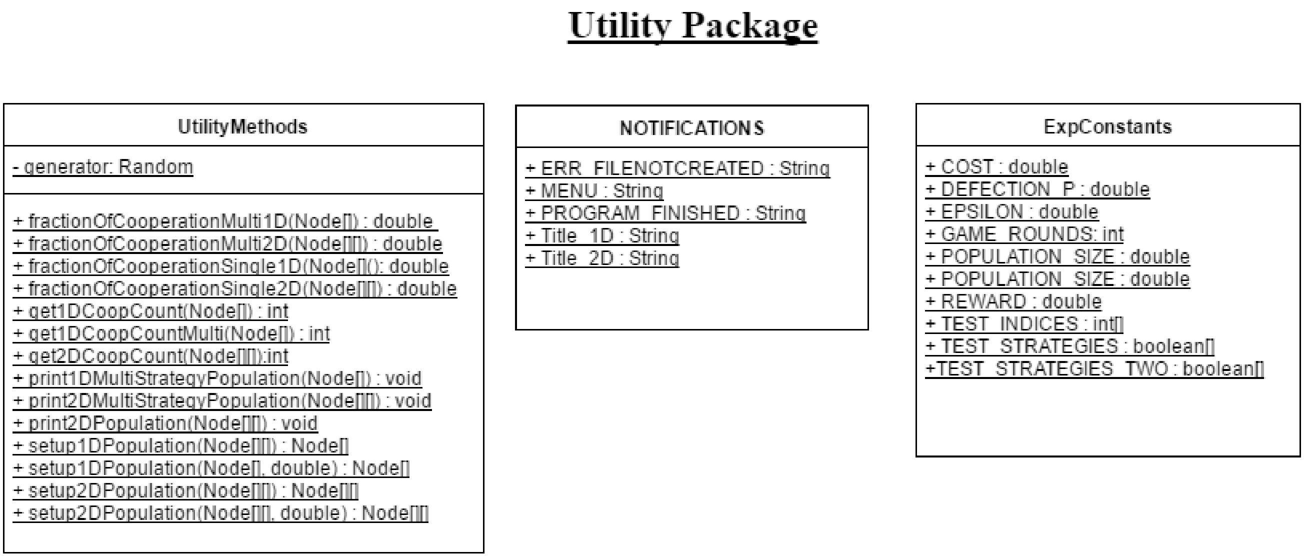

Figure 3.2: Class Diagrams for Utility.pkg

3.3.2 Game Model Package

Game models are contained in this package, these are the actual implementation of the evolutionary game theoretical models considered in this project. Figure 3.3 depicts the Class Diagrams of the Game Model Package

3.3.3 The Node Class

The node class constitutes a player in the simulation. Nodes were designed to store their fitness, maintain strategy lists, group tables, individual strategies and respective setters and getters. It also deals with setting up the strategy vectors to guarantee even 50 percent cooperation ratio. Individual nodes populate Rings and 2 dimensional arrays in the simulations. Figure 3.3 depicts the Class diagram of a Node object. The node is located in the GameModels.pkg. The class diagram for the node is shown in figure 3.4.

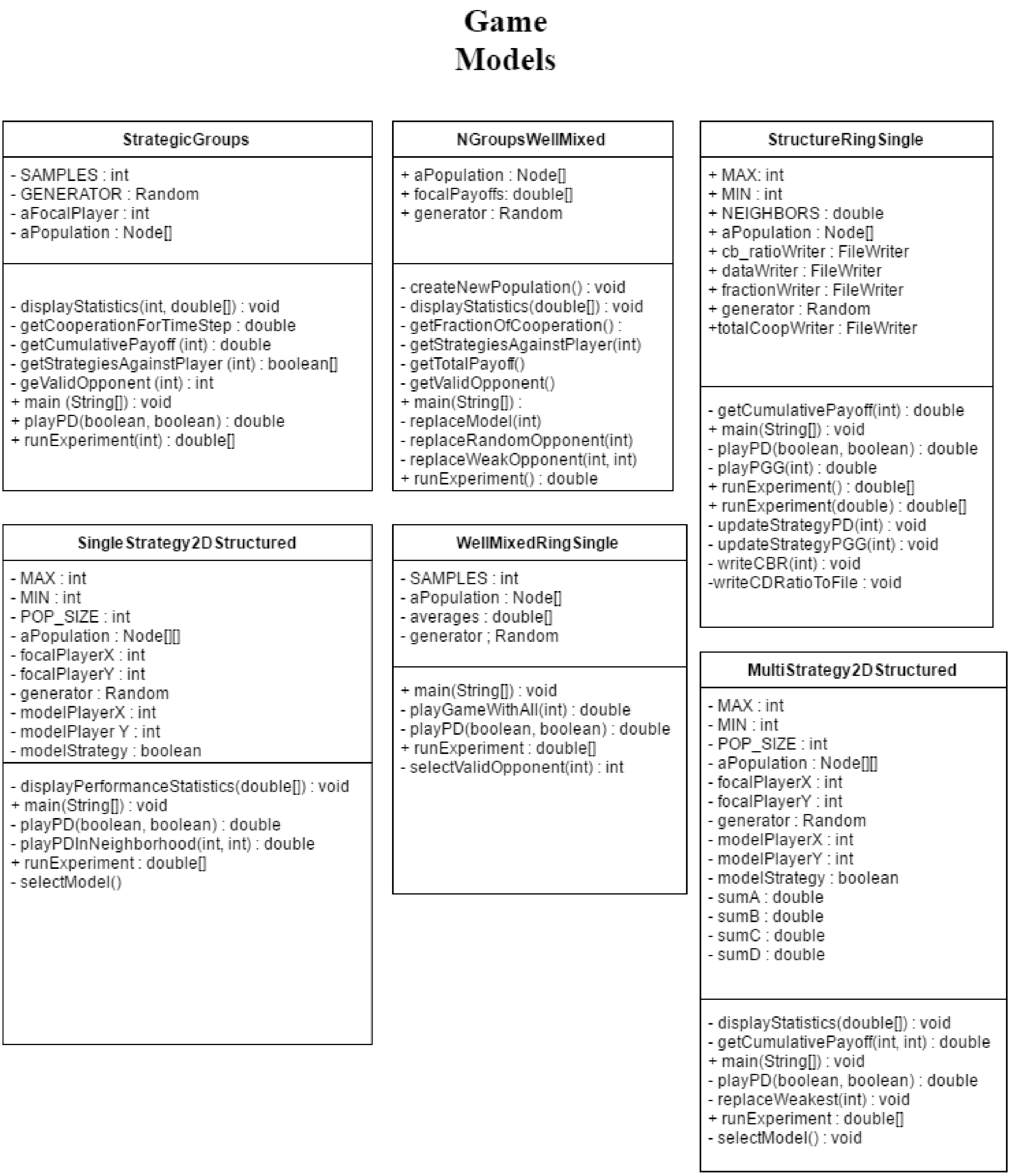

Figure 3.3: Class Diagrams for classes in GameModels.pkg

3.4 Testing

The iterated waterfall model was used for implementation and testing. Predictable test cases were executed, output was printed for each computation occurring in the game model and verified.

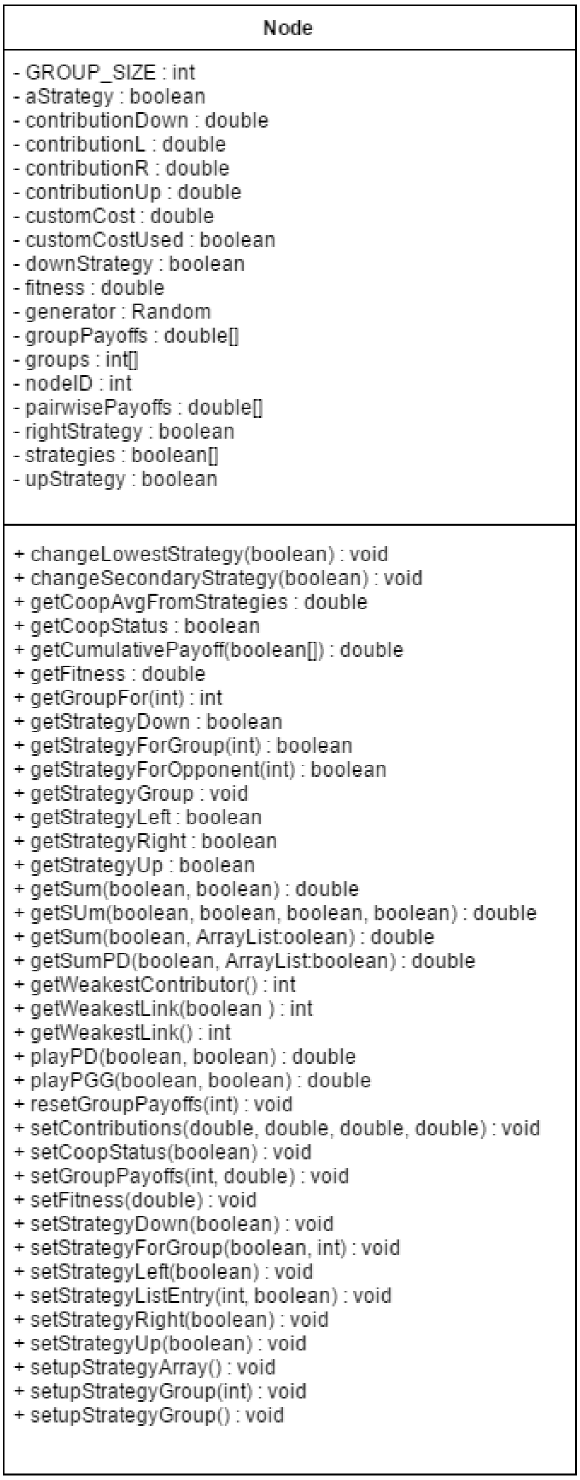

Figure 3.4: Class Diagrams the Node class

3.4.1 Instructions

The Maven Repository can be opened using an IDE such as NetBeans. NetBeans 8.2 is recommended. The appropriate game model can be selected in the GameModel.pkg. Each Model contains a main method. This is the program's main point of entry. The main methods contain Buffered File

Writers, file names can be altered inside main. Main can be rewritten to run the desired kind of experiment. Values such as size of the populations as well as the return values from the Game model can be set in ExpConstants.java under Utility.pkg. After execution of a game model, the data pertaining to the model is displayed in the NetBeans console. This is accompanied by a representation of the society in its final state, provided this is feasible. Each execution generates a CSV (Comma Seperated Value) file which contains the fraction of cooperation for each time step. Importing these into an environment such as MatLab, allows users plot graphs representing the data.

Chapter 4

Results

This chapter contains the key variables and why these are relevant when comparing the performance of the discussed models. The method subsection details the methods and the various parameters set to obtain them.

4.1 Methods

The discussion will look at the evolutionary stability of each model, the individual performance and highlight advantages, trade-offs and ecological validity of the game models.

This section identifies the key performance indicators and sets out how these were obtained. It itemises parameters and values supplied used in each individual experiment. Each discussed model was implemented using the Java Programming Language and used the standard iterated prisoner's dilemma game for a finite number of rounds. The Monte Carlo method was used to select random focal players. Model players and opponents are randomly selected adhering to the individual game-models' rules, as discussed in section 4. To produce comparable results of the game models' performance using single and multiple strategies, the following key data points require consideration.

For all experiments, the fraction of cooperation f_C was calculated which provides a way of quantifying the presence of cooperation in a population, inversely it quantifies defection. As a function over time, it can provide insights into how well a game model is performing in terms of maintaining or promoting cooperation. It can also grant insight into the interaction dynamics occurring within a population. Moreover, it provides a method of determining whether cooperation, if it exists, is evolutionary stable.

In single strategy games, the fraction of cooperation is defined as the number of cooperating nodes N_c divided by the size of the population N, N_c / N. In games using multiple strategies, an average local to the individual node is computed at first. In a scenario where a node maintains multiple strategies, we define this local quotient as the cooperative links maintained by a single node, divided by the number of total links. These values are then obtained for each node in the population and averaged using the population size as a divisor. In the special case in which a node in a well-mixed population associates a strategy with all its link, f_C was obtained by dividing the local co-operator links by the length of the strategy vector N-1. The fraction of cooperation is a dependent variable and is influenced by the probability to defect b.

The value of b is constant and denotes the defector payoff in the Prisoner's Dilemma game, it is $>> 1$. To obtain a measure for a model's resilience, values were adjusted for b within a range from 1-5, incremented at a rate 0.25. We observed whether these adjustments affected the overall performance, denoted by the final f_C obtained from an experiment for a given game model.

Depending on the game model we used population sizes of 10, 40, 100, 128, 256, 512, 1024. Some of these values, such as 40, were only used to verify the efficacy of the model's implementation, e.g. (multi strategy 2d). The varying population sizes formed the basis for an analysis of the effect the scale of the population had on f_C. This enables the drawing of conclusions as to whether a model is scale-variant.

To facilitate this, the program played 100,000 rounds of the iterated Prisoner's Dilemma game, this value was conditionally adjusted upward with values up to 1,000,000 for larger populations. Larger populations have been found to take longer to produce a stable final f_C value. This extra-step was only carried out if for sizes > 256 and if final f_C values were much higher than those of smaller populations and if this was confounded by a pattern showing a smaller rate of change in f_C values than observed in smaller populations.

To obtain data sets of higher quality, 1,000 samples for each experiment were sequentially executed. The f_C values recorded at each time-step of each sample were averaged. For each sample, a unique population with a f_C of 50 percent was created. This ensured that variety of possible initial configurations could be tested; this sample size was increased to 1,000 for larger populations to ensure that a greater variety of initial configurations could be tested.

For players of the multiple strategy game it was ensured that their initial strategy configuration was aligned so to produce the same initial f_C value. In the game model using multiple strategies on a square lattice, strategies were uniformly set to either cooperate or defect.

To increase the ecological validity of the data, noise was added in the form of misjudgements during the strategy update. For 10 percent of all updates, a player of the prisoner's dilemma carried out the strategy update for a random opponent, instead of the of the intended recipient.

To analyse the overall performance of the population, the payoffs and cumulative payoffs for single strategy and multi-strategy models respectively were summed. This represented the total payoff per population and was averaged through the population size, to obtain the average fitness per node. This way of quantifying performance is closely modelled after the GDP per capita model in economics.

Obtaining the respective payoffs per co-operator and payoff per defector, shows the individual performance of a player.

Finally, a hypothetical cognitive cost was considered, its aim is to reflect and quantify how cognitively demanding the maintenance of multiple strategies is. As implied, this only applied to multi-strategy models and is founded on the assumption that single strategy games model organisms with low or no visible intelligence, such as single-cell or multi-cellular organisms.

Values obtained for each data set were plotted using Matlab. Considered were the individual performance of each game-model, and the performance of a game-model at various sizes and values for b.

4.2 Results

4.2.1 Scalability

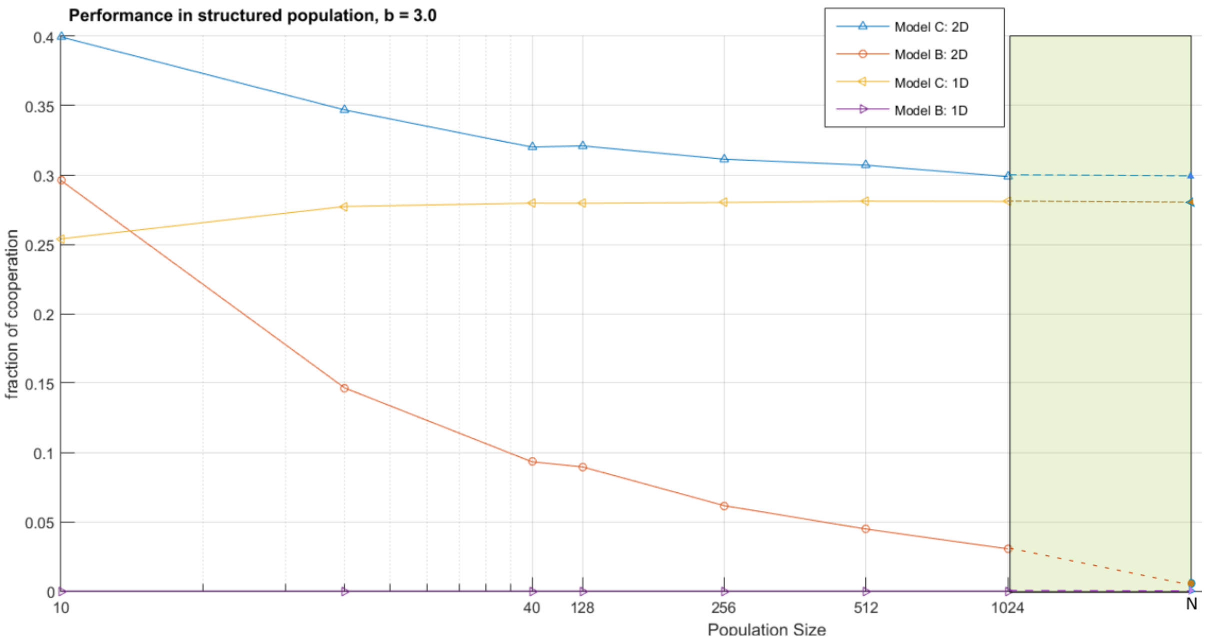

Figure 4.1: Scalability of asynchronous game models with single and multiple strategies (B, C) in 1 and 2D structured populations. Model C in 2D and 1D . Model B in 2D, Model B 1D. Green area shows projected performance. Shown are f_C averages from 1,000 samples contrasted to varying population sizes.

Figure 4.1 shows the performance of Models B and C in a structured population. The dynamic strategy model in 2D shows a 10 percent f_C decline over a 1,000 percent increase in size. The decrease occurs at a steady rate, as the population size is doubled. Applied in a 1-dimensional structured population, the f_C steadies at 0.28, the strategy is evolutionary stable.

The single strategy game in 2D exhibits a steady decline of cooperation as the number of players is increased. The model does not support cooperation in a 1 dimensional structure

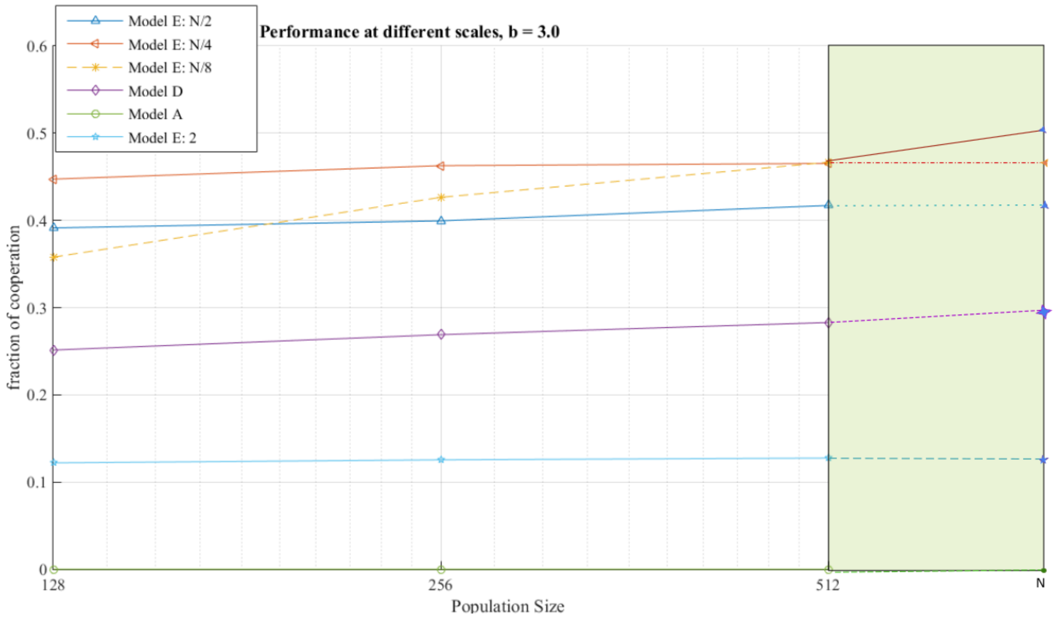

Figure 4.2: Scalability of asynchronous game models in a well-mixed population. Depicted are the single strategy game in the well mixed population, Model D using N-1 strategies and Model E with strategy groups N/2,. . . ,N/8 and 2 groups. Shaded area shows projected performance.

Figure 4.2 depicts the average fraction of cooperation realised for 1,000 samples for various population sizes ranging from 128 to 512. The shaded area shows the models' estimated future performance. The ordinary game in the well mixed population (Model A) results in the Prisoner's Dilemma, cooperation cannot be maintained. Model D with N-1 strategies maintains existing clusters of cooperation, is evolutionary stable and robust to errors. Splitting opposing players into two groups results in marginally lower, albeit stable levels of cooperation. Further dividing opponents shows a marginal increase in levels of cooperation maintained compared to Model D. Model E with N/2 groups maintain stable levels of cooperation roughly 15 percent higher than observed in Model D. N/4 exhibits the same evolutionary stability at a marginally higher level. N/8 shows an increase in f_C as the number of players increases.

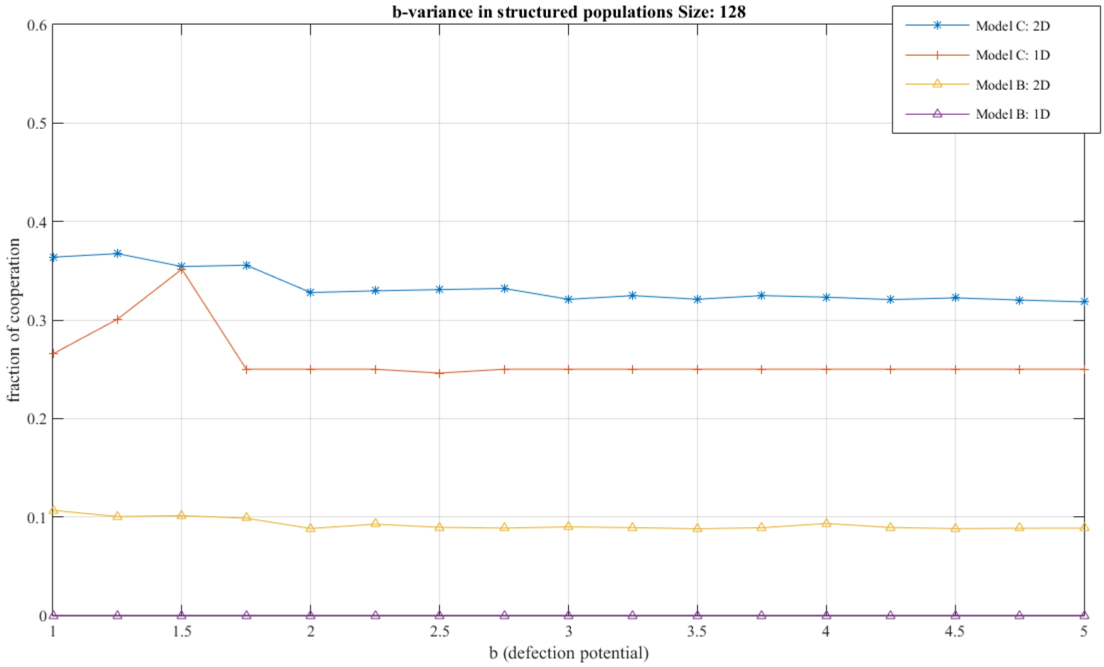

Figure 4.3: Fraction of cooperation for different values of b, ranging from 1-5 for different egt-models in a structured population.

4.2.2 b-variance

Figure 4.3 shows the simulation results for Models B and C, which operate in a structured population. Except for an anomalous spike in the 1 dimensional model using multiple different strategies against its opponents, all models maintain relatively stable levels of cooperation across all samples taken for the range of b. The single strategy game in a structured ring (Model B 1D) results in the Prisoner's Dilemma. In this case, the implemented mechanism fails to prevent cooperating players from being exploited. Models using different strategies against different opponents maintain higher levels of cooperation than the single strategy models.

Figure 4.4 shows the simulation results of strategies in well-mixed populations. Measures indicate whether a strategy is affected by an increase in the payoff defectors receive. Included are the single strategy game, Model D with N-1 strategies and the strategic group Model E with N/2,. . . N/8 and two strategy groups. The single strategy model exhibits the prisoner's dilemma for all b > 1. Model E with N/8 strategies exhibits a gradual decline in f_C per 0.25 b-increment. Model E with 2 strategy groups shows a growth in defection from

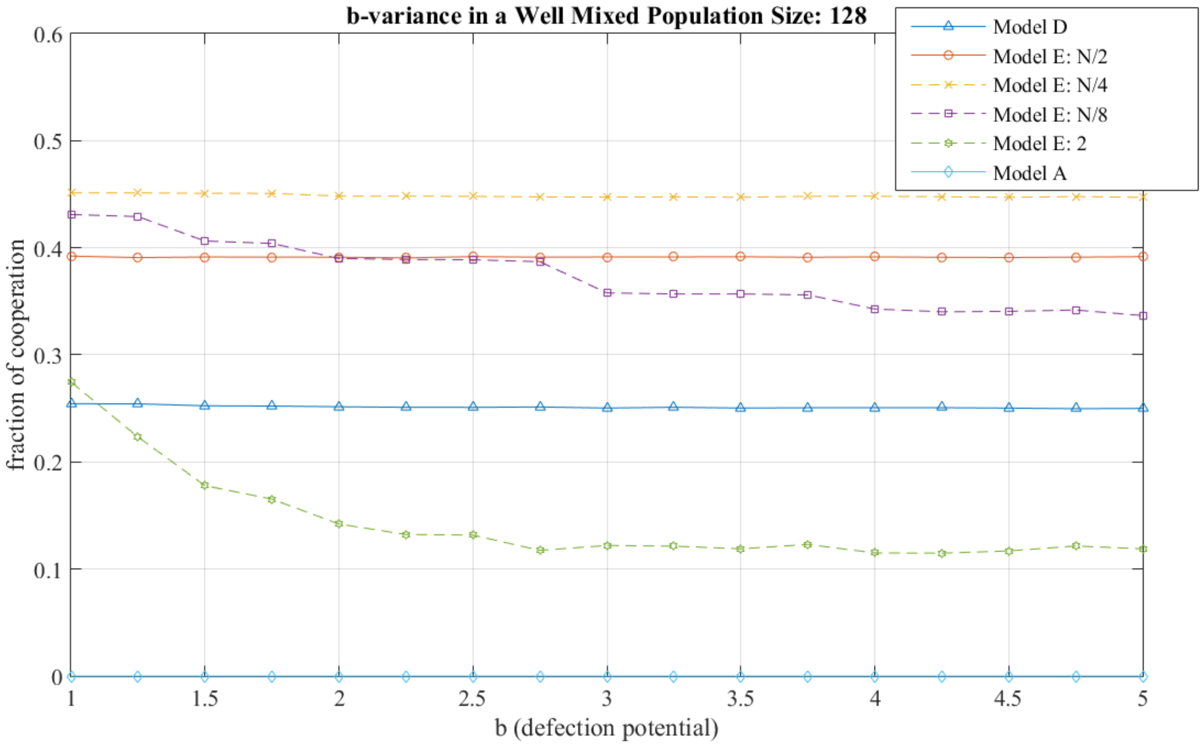

Figure 4.4: Fraction of cooperation for different values of b, ranging from 1-5 for different egt-models in a Well-mixed population.

b from 1 to 2.75. It fluctuates around 11 percent cooperation for remaining b values. The remaining game models exhibit robust levels of cooperation for all values of b measured.

4.3 Analysis

Observing the visual output produced by Models B and C confirms the formation of cooperative clusters as reported in other research. These clusters can be seen in Figure 4.5; their formation prevents cooperating players from being invaded by defectors. The single strategy model with players interacting in a 2-dimensional neighbourhood maintains marginal levels of cooperation, with 10 percent co-operators remaining shielded from defectors. The model is b-invariant, but not scale-free, resulting in a marked decline of cooperation as population size is increased. This is in contrast to the reported findings, which ascribe the model cooperative success at low levels of b. The model is clearly b-invariant but only manages to support marginal levels of cooperation. Its single-dimensional counterpart sees players interacting with their directly adjacent neighbours.

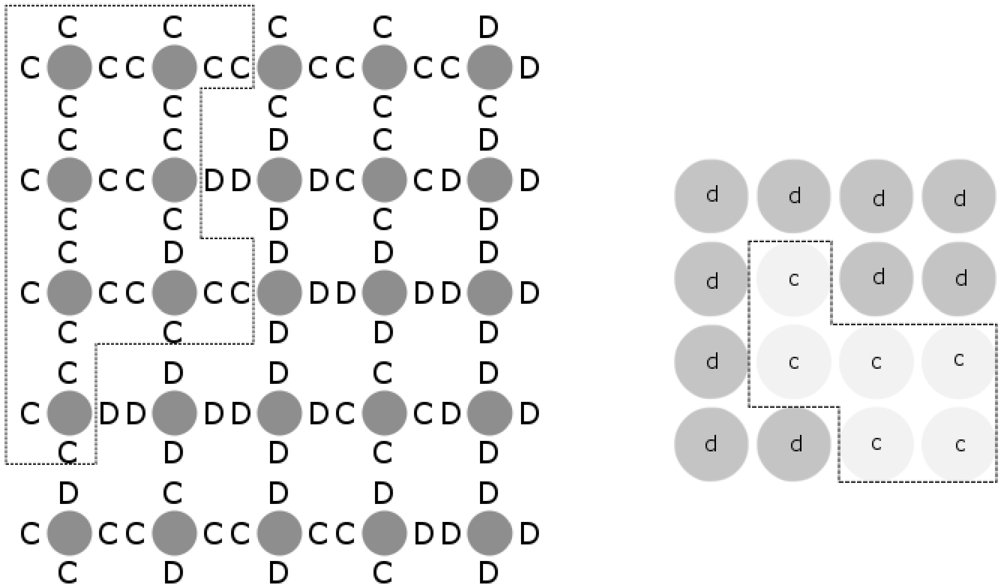

Figure 4.5: A representation of cooperative clusters as observed in the data sets.

Here the spatial mechanism fails to promote cooperation and results in the Prisoner's Dilemma. These observations allow the conclusion that the spatial mechanism, albeit somewhat efficient, does not guarantee the formation of cooperative clusters and is not evolutionary stable. In contrast to this, Model C's application of diverse strategies maintains higher levels of cooperation. This is a direct result from players' ability to neutralise encounters with defectors while maintaining links with other co-operators. The model applied to the ring achieves both higher levels of cooperation than its direct counterpart, as well as the single strategy model in the 2-dimensional domain. It maintains a steady level of cooperation for varying population sizes and shows the same characteristics even if the incentive for a player to defect is increased. This model in a 2-dimensional domain, maintains marginally higher levels of cooperation after initial C-D encounters have been neutralised. It is also largely b-invariant and scale-free. These findings verify the claims made in their paper of origin to an extent and are in support of the hypothesis formulated in this paper.

A general weakness this model displays is the lack of ecological validity. Nowak asserts that the spatial model is a good approximation of real-world interactions as it bears greater resemblance to real-life societal structures. This is not always the case, as the spatial models used in this project restrict the number of possible interactions to two and four, respectively. A more dynamic model with different levels of social connectivity, e.g. players exhibiting a varied number of connections, could provide a better approximation of real-life dynamics.

The model seeing users equipped with a single strategy for all their interactions (Model A) performs as expected in the Well-mixed population. The absence of protective mechanisms causes defectors to quickly dominate the entire population and results in the Prisoner's Dilemma for all players. The inclusion of social diversity in the asynchronous game model maintains existing (C,C) encounters, as is evidenced by the performance shown for Model D. This results in a decline of about 50 percent from the initial 50/50 configuration. However, the model maintains an f_C of 0.25 at the measured scales and is projected to be scale-free. Moreover, performance is unaffected by an increased incentive to defect. This leads to the conclusion that the model is evolutionary stable. It is worth noting that its study of origin focuses on the synchronous model, which claims to promote an increase in cooperation. The scope of this project only comprises results from models using asynchronous update.

As mentioned in the background section, the model aims to approximate human behaviour but it suffers from a range of drawbacks. The well-mixed population model is an interesting metaphor, but does not represent real-life social dynamics. Seeing each player connected to every other player and maintaining a discrete strategy for each interaction is not only a vast oversimplification of real-life systems, but also comes with a significant cognitive overhead. While theoretically feasible in small-scale networks, up-scaling such the population results in players having to memorise an increasing number of strategies, as well as account for a linearly increasing number of contributions, when deciding which opponent to carry out a strategy update for. Moreover, this is reflected by the significant computational overhead the simulation implementing this model has.

These drawbacks are addressed by the novel model proposed in this project. As previously discussed, the model incorporates a grouping mechanism combined with the social learning approach used by the evaluated models. Players in a well-mixed population segment N-1 opponents into a variety of smaller groups. The aim is to model evolutionary dynamics in a situation where a species exhibits lower cognitive abilities and therefore identifies opponents by groups.

Figures 4.2 and 4.4 identify this as Model E, it segments opponents into two strategy groups. Initially, it shows a steady decline in f_C , as the value for b is increased. This decrease tapers off at b=2.75 and continues to exhibit minor f_C fluctuations around 0.11. The model is the cognitively least expensive and a population using this still maintains a robust, albeit low f_C. Defection is clearly the dominant strategy, which is a result of players not being able to select and protect against individual free-riders that have invaded a group. It shares a similarity with the structured population.

Using $N/2$ groups sees the opponents split into a pairwise configuration. This approach yields higher, stable levels of cooperation than the previously discussed model. Contrary to Model D, a player now memorises pairwise-contributions to its fitness. This is less costly and enables the player to neutralise defectors more efficiently. A (C,(D,D)) encounter, where the inner brackets denote the strategies inside an opponent group, will invariably be replaced with (D,(D,D)). Further reducing the number of strategies to N/4 results in yet another minor f_C increase. Both methods appear scale-free and the mechanisms achieve to maintain their levels of performance over the spectrum of tested b-values. The behaviour of the model with N/8 strategies exhibits somewhat different behaviour. Its performance is less robust, as evidenced by the increase of defection that arises with increased defection potential. This may also be caused by the introduction of noise. The fraction of cooperation increases as the number of players is increased.

Chapter 5

Conclusions and Pestle Analysis

5.1 Discussion and future research

The analysis of the implemented models confirms this project's research question. It is evident from obtained metrics that game models applying social diversity outperform their single-strategy counterparts in the Well-mixed and spatial domains. The proposed model using strategic groups enhances this performance and offers a less costly alternative to the model using N-1 strategies. Research conducted in this project exhibits several shortcomings with respect to its thoroughness.

The original studies featured both asynchronous and synchronous update methods, with a clear emphasis on the latter. The synchronous update method sees players adjusting their strategies simultaneously, during an encounter. Therefore, it is more likely to produce high levels of cooperation; the asynchronous counterpart only maintains levels of cooperation. Its implementation and the consequent research of its performance would have been desirable.

With respect to the investigation of models in the well-mixed domain, tests for larger population sizes should have been performed. The algorithms' procedural implementation caused their execution to be enormously time consuming, some of them taking up to 70 hours. This may also have been the result of insufficient system memory. The algorithm could have been rendered more efficient by using threads, which concurrently simulate the behaviour of a multitude of populations. While the Java Programming language offers facilities for this, re-implementation in a faster performing language, such as C++, could have also reduced run-times.

The predicted f_C in Figures 4.1 and 4.2 have no scientific basis other than behaviour observed in the models. Conducting a mean-field analysis using a replicator equation would have provided a more robust insight into predicted future performance and could have rendered the sampling of larger population sizes superfluous. Mean field analysis was also considered in the studies which form the basis of this research project.

A more thorough analysis of the data was envisioned, but data sets obtained focused only on a very narrow aspect, namely the fraction of cooperation in relation to b-variance and population size.

Overall, the implementation cannot be considered user-friendly. Knowledge of high-level programming languages is a pre-requisite and the provided documentation is arguably too sparse. A straightforward user-interface, either in command line or with graphical components, would have improved this. Visualisation of the data was carried out with the MATLAB programming language, yet JAVA contains exhaustive graphic libraries. A simpler and more straight-forward way of presenting the data directly, could have been achieved.

For simplicity's sake several methods which should have logically been placed in the Utility Library were retained in the respective Game Model classes. They serve similar purposes, but due to variations in the game models' operations achieve this through different means. Object-Oriented languages generally provide the ability to use Polymorphism, and many methods implemented already make use of this. Ideally, this should have been applied to methods such as printData() as well.

The broad spectrum of multidisciplinary research into the field of Network Science and particularly the study of the emergence of cooperation made selection of appropriate sources difficult. Literature including topics such as cellular automata[3], evolutionary biology, biological mathematics and Phyiscs were considered, however conceptual understanding was difficult to achieve in some cases, due to limited background knowledge.

Several computed data-sets were not used and some methods featuring a more diverse range of update strategies were implemented, but did not form part of this project. These could be considered in future research.

Implemented Game models use symmetric payoffs, meaning that achieved payoffs are the same for all players. Some researchers proposed and studied the effects of using asymmetric payoffs, future research could incorporate these [9].

Additionally, used models could be fitted with a reputation system as described in the research discussed in the Background chapter of this paper.

5.2 Pestle

The models contained in this projects serve as an abstraction of real-life dynamics. As is common in theoretical systems modelling real-life processes, they are highly reductionist. Results and predictions should be taken as an indicator, but should not be used to explain real-life dynamics, as these are arguably far more complicated and cannot simply be explained through a singular mathematical model. Applying such models to solve problems in technology may not provide adequate solutions, for example to solve problems such as resource-starvation. Moreover, the models implemented rely on symmetric information, when in real life, flow of information is often asymmetric[16].

5.3 Conclusion

This project replicated and compared evolutionary game theoretical models using social diversity and single strategy approaches in well-mixed and structured population models. A novel approach founded on the principle of social-diversity in well-mixed populations, envisioned to model cooperative dynamics in species with reduced cognitive abilities, was proposed. The novel approach successfully encouraged cooperation compared to models with single strategies.

Appendix A

References

[1] R. Axelrod, ‘Effective choice in the prisoner’s dilemma’, *The Journal of Conflict Resolution*, vol. 24(1), pp. 3–25, Mar. 1980 (p. 5).

[2] ——, *The Evolution of Cooperation*. New York, NY: Basic Books, 1984 (p. 6).

[3] E. Codd, *Cellular Automata*. Cambridge, MA: Academic Press, 1968 (p. 37).

[4] C. Darwin, *On The Origin Of Species*. New York, NY: Sheba Blake Publishing, 2014 (p. 3).

[5] S. H., ‘The survival of the fittest’, *Nature*, vol. 5(118), pp. 263–264, Feb. 1872 (p. 3).

[6] M. Haahr. (). Introduction to randomness and random numbers, [Online]. Available: `https://www.random.org/randomness/` (visited on 15th Mar. 2017) (p. 6).

[7] G. Hardin, ‘The tragedy of the commons’, *Science*, vol. 162, pp. 1243–1248, Dec. 1968 (p. 3).

[8] C. Hauert. (2005). Public goods games: Well-mixed populations, Universitaet Wien, [Online]. Available: `https://www.univie.ac.at/virtuallabs/PublicGoods/mixed.html` (p. 8).

[9] A. McAvoy and C. Hauert, ‘Asymmetric evolutionary games’, *PLoS Computational Biology*, vol. 11, no. 8, 2015. DOI: `10.1371/journal.pcbi.1004349` (p. 37).

[10] Y. Morvanand and C. O’Sullivan, ‘Slacer: A self-organizing protocol for coordination in peer-to-peer networks’, *IEEE Intelligent Systems*, vol. 21, no. 2, pp. 29–35, Apr. 2006, 2006 (p. 1).

[11] J. von Neumann and O. Morgenstern, *Theory of Games and Economics Behavior*. Princeton, NJ: Princeton University Press, 1944 (p. 4).

[12] M. Nowak, ‘Five rules for the evolution of cooperation’, *Science*, vol. 314, no. 5805, pp. 1560–1563, 2006. DOI: 10.1126/science.1133755 (p. 4).

[13] A. owak and M. May, ‘Evolutionary games and spatial chaos’, *Nature*, vol. 359, pp. 826–829, Oct. 1992 (p. 7).

[14] *Introduction to Monte Carlo simulation*, vol. 24(1), IEEE, Dec. 2008. DOI: 10.1109/WSC.2008.4736059 (p. 6).

[15] F. Santos, M. Santos and J. M. Pacheco, ‘Social diversity promotes the emergence of cooperation in public goods games’, *Nature*, vol. 454, pp. 1–12, Jul. 2009 (p. 9).

[16] J. Stiglitz, ‘Information and the change in the paradigm in economics’, *The American Economic Review*, vol. 92, no. 3, pp. 460–501, 2002 (p. 37).

[17] L. Wardil and J. K. L. da Silva, ‘Adoption of simultaneous different strategies against different opponents enhances cooperation’, *EPL (Europhysics Letters)*, vol. 86, no. 3, p. 38 001, 2009. [Online]. Available: http://stacks.iop.org/0295-5075/86/i=3/a=38001 (p. 9).

[18] L. Wardil and J. da Silva, ‘Distinguishing the opponents promotes cooperation in well-mixed populations’, *Physical Review*, vol. find, pp. 036115-1 - 0356115-7, Mar. 2010 (p. 10).

[19] E. Weisstein. (1987). Von neumann neighborhood, Wolfram Mathworld, [Online]. Available: http : / / mathworld . wolfram . com / vonNeumannNeighborhood.html (visited on 15th Mar. 2017) (p. 7).

[20] C. Xia, S. Ding, C. Wang, J. Wang and Z. Chen, ‘Risk analysis and enhacement of cooperation yielded by the individual reputation in the spatial public goods game’, *IEEE Systems Journal*, vol. 1932-8184, pp. 1–10, Apr. 2016 (p. 7).

Appendix B

Other Resources

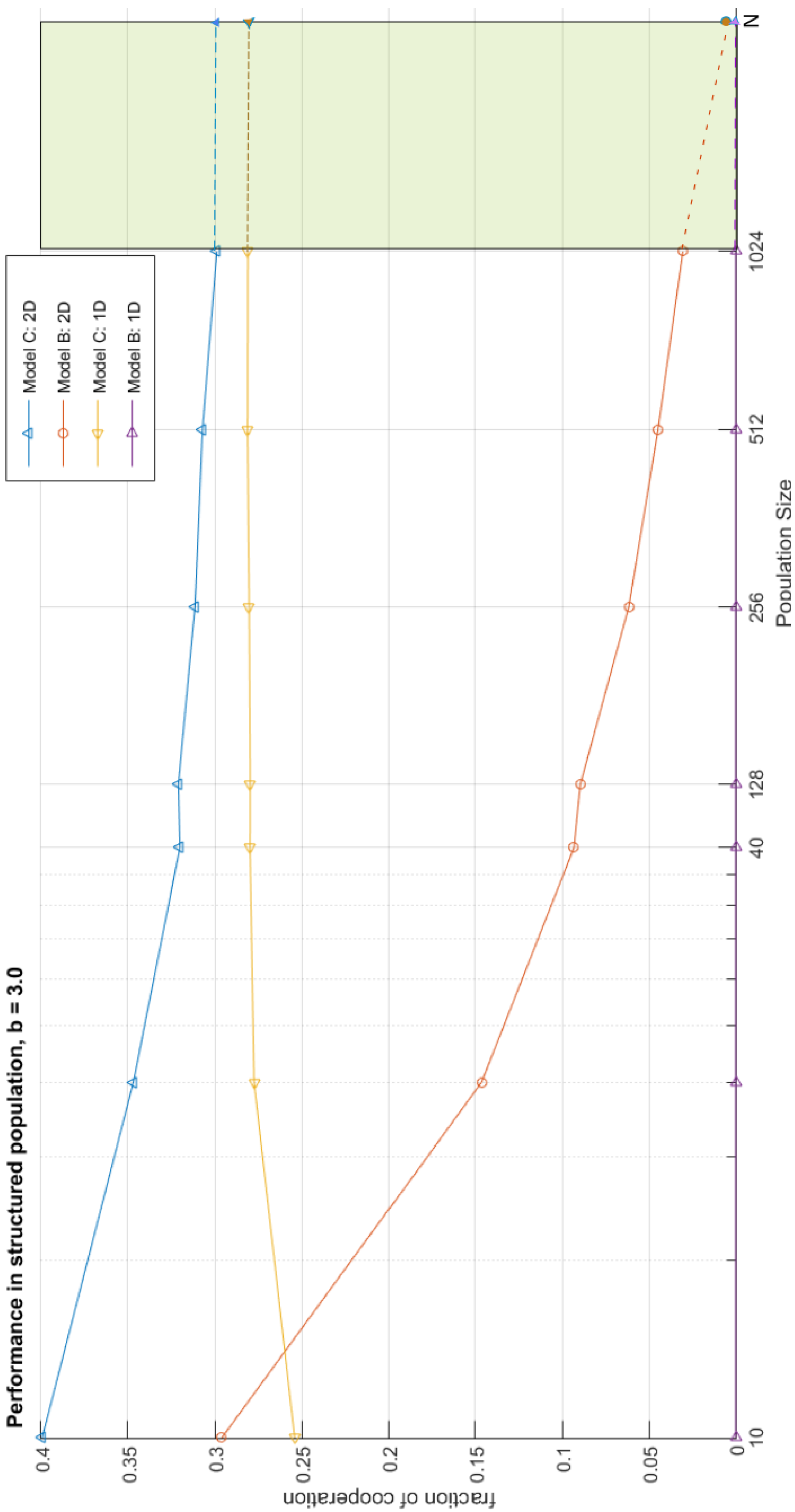

Figure B.1: Large version of graph showing performance to scale in structred populations

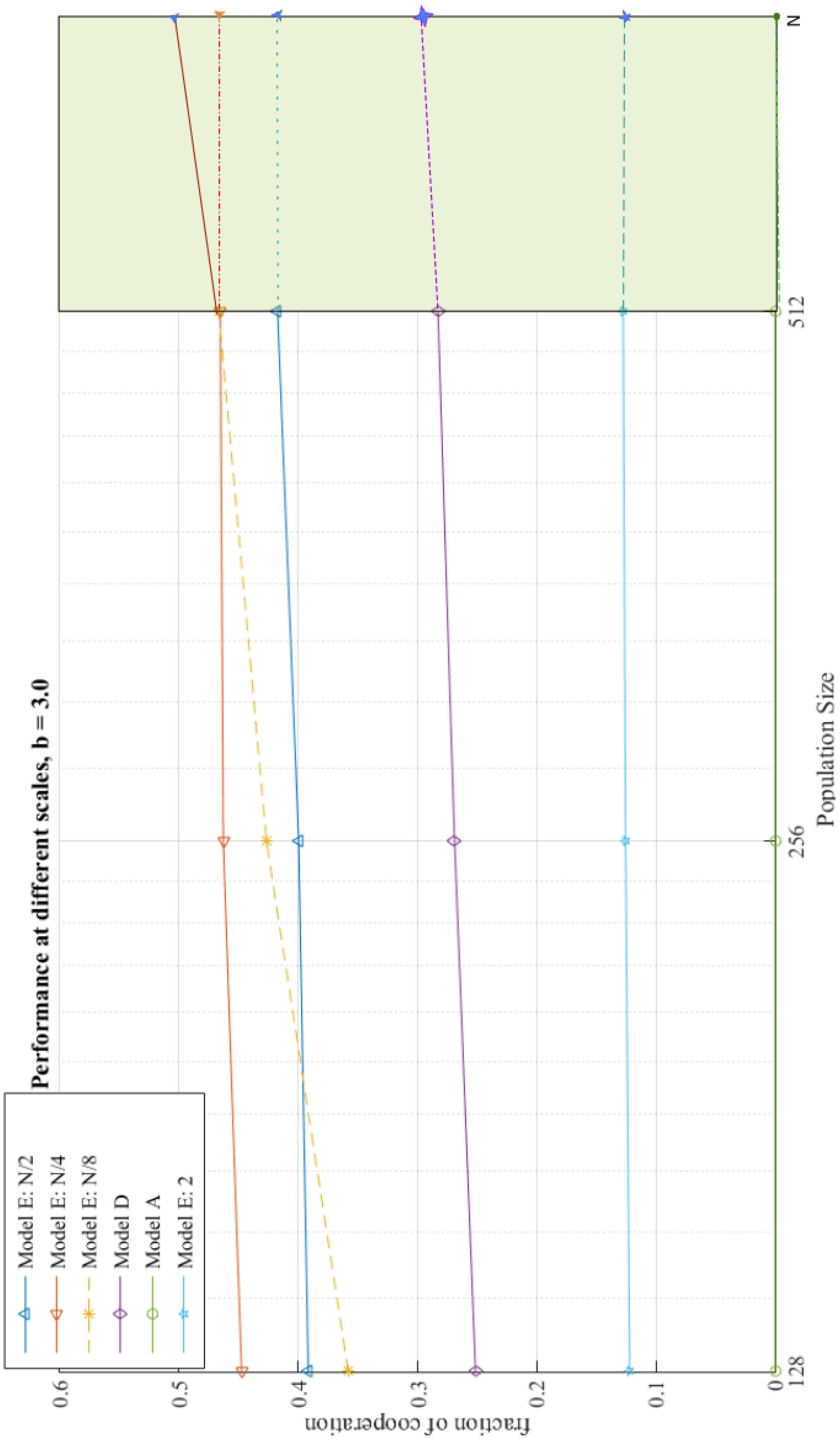

Figure B.2: Large version of graph showing performance to scale in well-mixed populations

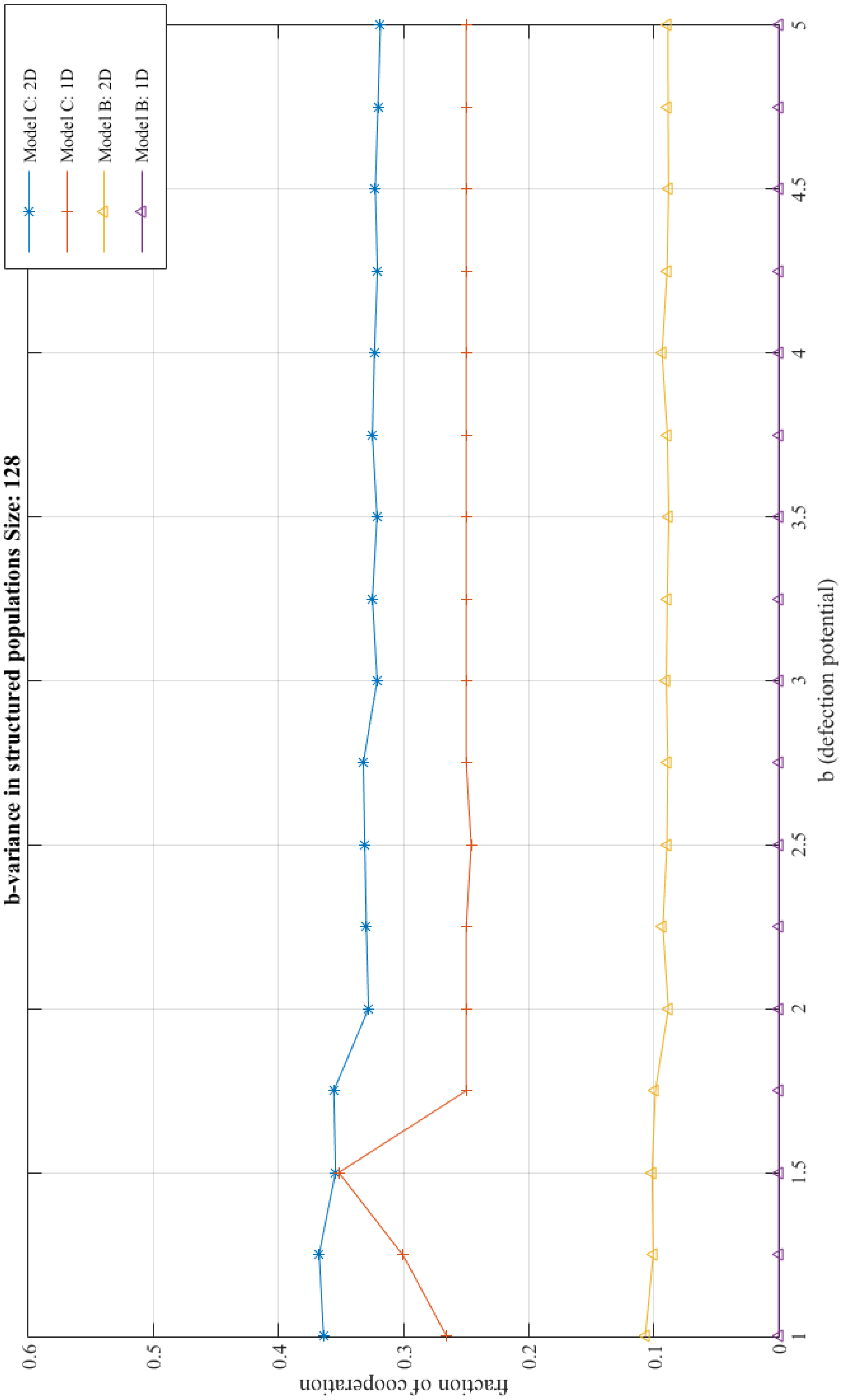

Figure B.3: Large version of graph showing performance to defection potential in structured populations

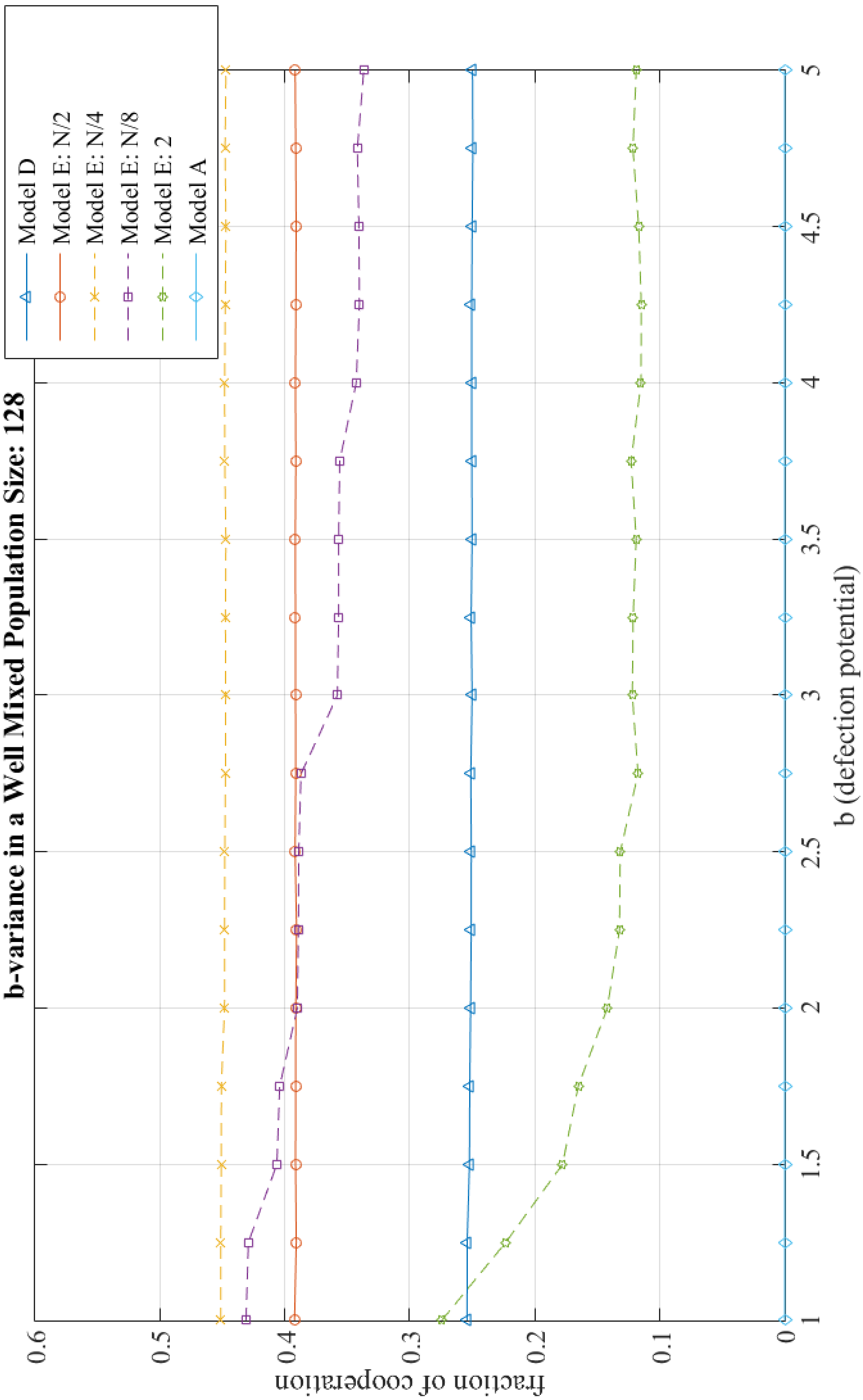

Figure B.4: Large version of graph showing performance to defection potential in well-mixed populations

Appendix C

Project Poster

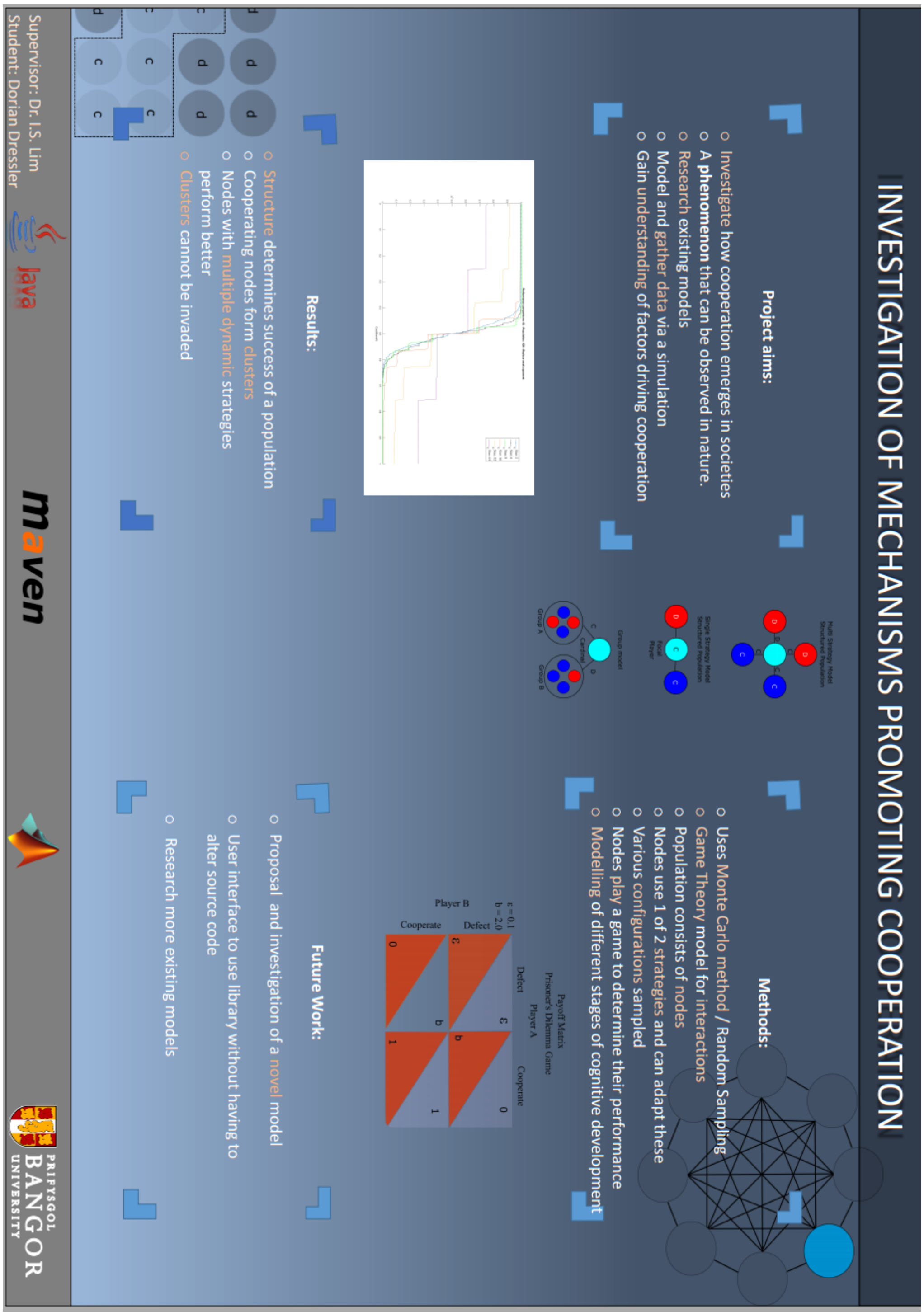

Figure C.1: Project Poster