

Modelling predator-prey interactions with cellular automata

Dirk J. Human

Research assignment presented in partial fulfilment of the requirements for the degree of
BComHons (Operations Research)
at the Department of Logistics, Stellenbosch University

Abstract

In systems biology, one of the most well-known dynamics is an interaction between multiple species, known as predator-prey interaction, in which some species act as predators and others as prey. The modelling of this interaction is gaining ground in fields such as conservation ecology, not least because it is significantly more cost-effective than traditional experimental observation. Modelling also poses no actual threat to a real-life system, which may be made unstable by external influences. Since the first formal mathematical model was introduced by Lotka and Volterra in the mid-1920s, several differential equation-based models have been developed to describe predator-prey interaction. However, a key disadvantage in most of these models is that it is difficult to add additional complexities while retaining the ability to analyse the results in a meaningful way. Also, although these models give a good macroscopic view of a system, they are not well-suited to observing microscopic interactions. In recent years, with the increasing computational power brought by computers, predator-prey interactions have been modelled successfully using simulation models. This has made it possible to study more of the local interactions between individuals. A discrete simulation model for modelling two-species predator-prey interaction, known as a cellular automaton (CA), is presented in this study. The model specifically investigates the effect of the basic survival instincts known as predator pursuit and prey evasion. Individuals within the respective populations are given the ability to survey their surroundings and behave in such a way as to mimic the evasion of a threat (prey evading predators) or the active seeking of dense energy sources (predators pursuing prey). The models developed here all compare favourably with existing mathematical models in that they display similar reactions given equivalent parameters, while being simple to implement.

Uittreksel

Een van die bekendste dinamika in stelselbiologie is 'n interaksie tussen veelvuldige spesies wat bekend staan as roofdier-prooi-interaksie, waarin party spesies as roofdiere optree, en ander as prooi. Die modellering van hierdie interaksie is besig om veld te wen op gebiede soos bewaringsekologie, onder andere omdat dit aansienlik meer koste-effektief as tradisionele eksperimentele waarneming is. Modellering hou ook nie enige aktiewe bedreiging in vir 'n werklike natuurlike stelsel wat deur eksterne invloede onstabiel gemaak kan word nie. Sedert die eerste formele wiskundige model in die 1920's deur Lotka en Volterra bekendgestel is, is verskeie differensiaalvergelyking-gebaseerde modelle reeds ontwikkel met die doel om roofdier-prooi-interaksie te beskryf. 'n Belangrike nadeel aan die meeste van hierdie modelle is egter dat dit moeilik is om addisionele kompleksiteite by te voeg en steeds in staat te wees om die resulatae op 'n betekenisvolle manier te analyseer. Alhoewel hierdie modelle 'n goeie makroskopiese oorsig van 'n stelsel kan gee, is hulle nie geskik vir die waarneming van mikroskopiese interaksies nie. Met die onlangse toename in berekeningskrag wat deur rekenaars meegebring is, kon roofdier-prooi-interaksies suksesvol gemodelleer word deur simulasiemodelle te gebruik. Dít het dit moontlik gemaak om meer van die plaaslike interaksies tussen individue te bestudeer. 'n Diskrete simulasiemodel vir die modellering van tweespesie-roofdier-prooi-interaksie, wat bekend staan as 'n sellulêre outomaat (CA), word in hierdie studie voorgestel. Die model ondersoek spesifiek die effek van die basiese oorlewingsinstinkte genaamd roofdierontduiking en prooi-agtervolging. Individue binne die onderskeie bevolkings word die vermoë gegee om hulle omgewing te beskou en een van twee gedragsvorme na te boots: bedreigingsontduiking (prooi wat roofdier ontduik) en die soek tog na digte energiebronne (roofdier wat prooi jaag). Die modelle wat hier ontwikkel is, vergelyk goed met bestaande wiskundige modelle deurdat hulle, gegewe ekwivalente parameters, soortgelyke reaksies toon terwyl hulle ook eenvoudig is om te implementeer.

Table of Contents

List of Figures	vii
List of Tables	ix
List of Algorithms	xi
1 Introduction	1
1.1 Background	1
1.2 Problem description	2
1.3 Objectives and scope	2
2 Literature review	5
2.1 Predator-prey interaction model	5
2.1.1 The Lotka–Volterra equations: An ODE approach	5
2.1.2 Diffusion: a PDE approach	6
2.2 Cellular automata	7
2.2.1 Definition	7
2.2.2 A brief history	7
2.2.3 Applications in complex systems	8
2.3 Spatial artificial life models	9
3 Methodology	11
3.1 Cellular automaton	11
3.1.1 Geometry	11
3.1.2 Rules	12
3.2 Predator-prey interactions	13
3.3 The simple model	13
3.3.1 Assumptions and geometry	13
3.3.2 Rules	14
3.3.3 Model validation	15

3.4	Predator pursuit and prey evasion	16
3.4.1	Assumptions and geometry	16
3.4.2	Rules	18
3.4.3	Model validation	21
3.5	Density sensitivity and age adjusted mortality	22
3.5.1	Assumptions and geometry	22
3.5.2	Rules	23
3.5.3	Model validation	25
4	Results	27
4.1	Model implementation	27
4.2	Parameter testing	27
4.2.1	Fixed parameters and configuration	27
4.2.2	Simple model	28
4.2.3	PPPE model	32
4.2.4	DSAM model	36
4.3	Comparisons with existing models and array size	41
5	Conclusion	43
5.1	Summary	43
5.2	Contributions to literature	44
5.3	Future research	44
List of References		49
A	Results tables	51
B	Contents of the disk accompanying the project	53

List of Figures

2.1	Neighbourhood types	7
3.1	Cell shapes in a CA.	11
3.2	Game of Life example	12
3.3	Probabilities of prey birth and survival in the simple model.	15
3.4	Simple model single species validation.	16
3.5	Population dynamics of the simple model.	16
3.6	View cone of a cell	18
3.7	PPPE model single species validation.	22
3.8	Population dynamics of the PPPE model.	22
3.9	Density sensitivity comparison in the DSAM model.	23
3.10	Predator death as a linear function.	24
3.11	Prey dynamics in the single species DSAM models.	25
3.12	Population dynamics of the DSAM model.	26
4.1	Effect of parameter variation on the prey population in the simple model.	29
4.2	Effect of parameter variation on the predator population in the simple model.	30
4.3	Population dynamics for the simple model.	31
4.4	Snapshots of the simple model output.	32
4.5	Effect of parameter variation on the prey population in the PPPE model.	33
4.6	Effect of parameter variation on the prey population in the PPPE model.	34
4.7	Population dynamics in the PPPE model.	35
4.8	Snapshots of the PPPE model output.	36
4.9	Population dynamics in the PPPE model given the base case.	36
4.10	Effect of parameter variation on the prey population in the DSAM model	37
4.11	Effect of parameter variation on the predator population in the DSAM model.	39
4.12	Population dynamics in the DSAM model.	40
4.13	Snapshots of the cDSAM model output.	40
4.14	Snapshots of the base case cDSAM model output.	41

4.15	Defensive spiral emergence	41
4.16	Extinction by broken spiral.	42
4.17	Defensive spiral emergence	42
4.18	Unbroken ring in the 100×100 cDSAM model	42
4.19	Unbroken ring in the 50×50 cDSAM model	42

List of Tables

4.1	Comparison of mean population over five runs of the simple model.	28
4.2	Predator extinctions by parameter for the simple model.	31
4.3	r_v and the prey population in the PPPE model.	33
4.4	r_v and the predator population in the PPPE model.	34
4.5	Predator extinctions by parameter for the PPPE model.	35
4.6	α to parameter value mapping	37
4.7	r_d and the prey population.	38
4.8	r_v and the prey population for the DSAM models	38
4.9	r_d and the predator population.	39
4.10	r_v and the predator population for the DSAM models	39
A.1	Mean population sizes in the simple model.	51
A.2	Mean population sizes in the PPPE model.	51
A.3	Mean population sizes in the cDSAM model.	52
A.4	Mean population sizes in the eDSAM model.	52

List of Algorithms

3.1	Simple predator-prey rules	14
3.2	Feeding phase of the PPPE	18
3.3	Reproduction phase of the PPPE	19
3.4	Movement phase of the PPPE	20
3.5	Prey density	24

CHAPTER 1

Introduction

Imagine a pasture, only a couple of hectares large, surrounded by an impervious fence stretching too far below the ground to burrow under and too high to bound over. This pasture represents a type of closed system. Two interacting species may be added to this pasture, like foxes and rabbits, for example. The rabbit, as nature intended, would happily go forth and multiply unbounded were there no foxes. Unfortunately, the fox has developed a taste for rabbit over the millennia and would just as happily eat the rabbits, and then go forth and multiply.

If we were to count the number of foxes and rabbits in the pasture every day, we start to notice a trend in the numbers: within a closed system, the larger the fox population becomes, the more rabbits they will consume leading to a reduction in the rabbit population. This will in turn lead to the fox population dying off due to a lack of food, leading to reduced predation pressure on the rabbits allowing for growth in their population. Ultimately this leads us back to a growing fox population. This example illustrates the fundamental idea behind a simple predator-prey interaction model [19].

1.1 Background

In systems biology, one of the most well-known dynamics is the interaction between multiple species where some act as predators and the others as prey. While the above mentioned example is a simple representation of a predator-prey interaction, the model has its use in conservation biology [27, 40], especially where new predators are being introduced to an ecosystem. Using mathematical or simulation models is far more cost-effective than experimental observation with the additional benefit of not causing any possible instability to an ecosystem in question, which may lead to the extinction of one or more species.

A closed system with an arbitrary amount of interacting species can be modelled effectively by a system of ordinary differential equations, such as the Lotka-Volterra equations [26, 37]. This type of model may give insight into the macroscopic nature of the system. To allow for a microscopic view of the system dynamics, such as dispersal and pattern formation, a spatially explicit model is required. For example, a system of partial differential equations (PDEs) may be used to model dispersal in the form of diffusion and subsequent pattern formation [21].

PDEs, although effective, are solved numerically for many different times and positions in a spatial domain, making it computationally expensive. An alternative, as mentioned by Hawick

& Scogings [17], is a model of explicit simulation using agent-based animat artificial life models. The animat model is an agent-based approach to simulation that uses discretely modelled individuals with a set of attributes, such as age and hunger, and a set of rules that determines their behaviour. Animat models have been applied to some ecological and biological models [9, 12], including the predator-prey model [18]. These models can be less computationally expensive than PDEs, however, they represent a form of artificial intelligence and may require multiple runs to avoid “biased” decisions [17].

Another alternative to using PDEs, and closely related to the animat, is the cellular automaton (CA). A CA is a set of cells that have a finite amount of possible states [44]. The model progresses in discrete time steps and the new state of any given cell is determined by the state of that cell and the state(s) of its defined neighbour(s) in the previous time step [41]. A CA may be considered a form of animat with a significantly smaller set of parameters when it comes to an individual’s behaviour [17].

1.2 Problem description

Although investigation into practical uses for cellular automata (CAs) have been prevalent since the mid 1980s, applications of CAs to biological systems have been tested as far back as the 1950s [45]. Its use for conservation purposes is not a new idea, and it has been applied to study, for example, the effect of invasive species on an ecosystem [3]. The implementation of various predator-prey models on CAs are also available [4, 17, 29], including some exploring the configuration of the CA and its effects on the predator-prey model itself [6]. The many parameters of a CA has, however, limited these studies to the analysis one or two of the available parameters, leaving much to be explored still.

This project will study the effect of predator pursuit and prey evasion on the system stability by means of extended neighbourhood awareness in the CA. Predator pursuit represents a predator’s desire to move towards areas of higher prey density and prey evasion the desire for prey to move away from any nearby predators. Many CAs have neighbourhoods that are limited to the cells bordering a cell in question, though it is possible to extend the radius of the neighbourhood or define a completely new neighbourhood.

1.3 Objectives and scope

The CA will be limited to a two-species system with one species preying on the other. With the exception of the DSAM model presented in §3.5, there will be no limit to the systems’ carrying capacity for prey apart from its size. The populations for both species will be homogeneous with the exception of age, therefore all parameters and their values will be the same for all individuals within the respective populations. Any model with explicit movement calculations will be limited to moving in a Von Neumann neighbourhood to limit the amount of computational time required.

The primary objectives for this project are:

- I Perform a literature review of the mathematical models developed for predator-prey interactions, specifically on artificial intelligence models.

- II Implement variations of some of the existing predator-prey models available and compare results obtained with those in the literature.
- III Modify these models by adding various complexities in their rules, including extended neighbourhood awareness and movement as to replicate predator pursuit and prey evasion and some added attributes such as age and density sensitivity.
- IV Investigate the stability of the new models and the effects of various parameter ranges and compare the results in the models developed with those in literature.
- V Identify any weaknesses or limitations in these models to aid in future research.

CHAPTER 2

Literature review

Mathematical modelling of the predator-prey interaction has its roots in independent observations by Lotka and Volterra in the early 20th century [26, 37]. Since then, many mathematicians, biologists and computer scientists have built on the model they developed, or created entirely new ones, often exploiting the raw calculation power brought by the computer. The development of the predator-prey model is discussed in §2.1, focussing on mathematical models such as ordinary differential equations (ODEs) in §2.1.1 and partial differential equations (PDEs) in §2.1.2. A brief history and description of CAs and their applications will be presented in §2.2 and the development of agent-based simulation models will be explored in §2.3.

2.1 Predator-prey interaction model

Predator-prey interactions would occur in any system with multiple species where some prey on others as a source of energy. Although these interactions may have been observed since the dawn of man, attempting to mathematically model predator-prey interactions is a recent development with the first known breakthrough only occurring in the 1920s.

2.1.1 The Lotka–Volterra equations: An ODE approach

In the 1830s, Belgian mathematician Pierre François Verhulst was inspired by *An Essay on the Principle of Population* by English scholar Thomas Robert Malthus [33]. Malthus claimed a population of size N would grow unbounded by some growth constant r . Verhulst disagreed with this, arguing that the growth of the population would be limited by some carrying capacity K , be it physical space or available resources [1]. He adapted Malthus' equation to

$$\frac{dN}{dt} = rN\left(1 - \frac{N}{K}\right), \quad (2.1)$$

commonly known as the logistic function or Malthus–Verhulst equation [36]. In 1910, American mathematician and chemist Alfred Lotka used the logistic function to model a set of consecutive chemical reactions that demonstrated damped oscillations before reaching equilibrium [24]. He adapted this model in 1920 to illustrate the oscillatory nature of certain reactions that seemed to reach an equilibrium due to mass action. He modelled it as

$$\begin{aligned} \frac{dN}{dt} &= rN - cNP \\ \frac{dP}{dt} &= bNP - mP \end{aligned}$$

where r denotes the growth rate of reagent N , c the reduction rate of N , b the growth rate of product P and m the reduction rate of P [25].

At the same time, Italian biologist Umberto D'Ancona noted that the proportion of undesirable selachians (predatory fish) caught off ports in Italy increased significantly during and immediately following the First World War, when commercial fishing was severely curbed. The opposite was also true, as the abundance of desirable food fish showed a notable decrease at the same time. He realised that the perturbation in both groups of species had to be related to each other. The problem was presented to Italian mathematician, Vito Volterra, whom D'Ancona hoped would be able to mathematically model this phenomenon [22].

In 1925 and 1926, Lotka and Volterra independently published the same equations for two different applications. Lotka based his on a herbivore eating plant material [26], and Volterra on the fish species of the Adriatic sea [37]. The equations

$$\frac{dN}{dt} = rN - cNP \quad (2.2)$$

$$\frac{dP}{dt} = bNP - mP, \quad (2.3)$$

where N denotes the number of prey, P denotes the number of predators, r and b denote growth rate coefficients and c and m denote death rate coefficients for prey and predators, respectively, would subsequently be known as the Lotka–Volterra equations. By setting the equations (2.2) and (2.3) equal to 0, population equilibria may be obtained as $\{N = 0, P = 0\}$ and $\{N = \frac{r}{c}, P = \frac{b}{m}\}$, representing either extinction of both species, or the point around which periodic oscillations would occur as observed by Lotka [24, 25]. Since the publication of the Lotka–Volterra equations, most investigations into predator–prey interactions utilised the Lotka–Volterra equations or some variation thereof.

2.1.2 Diffusion: a PDE approach

One of the greatest drawbacks in the Lotka–Volterra model is its simplicity, as pointed out by Gause [15], who argued that interaction between predator and prey was not a linear function, but some saturating function. This led research efforts to shift from ODEs to using existing systems of PDEs in the spatial domain.

The non-linear reaction–diffusion equation is commonly used to model systems of interacting components. The general form of the equation is

$$\frac{\partial \mathbf{u}}{\partial t} = f(\mathbf{u}) + \mathbf{D} \nabla^2 \mathbf{u}, \quad (2.4)$$

where $\mathbf{u} = \mathbf{u}(r, t)$ denotes the vector of concentration variables at position r and time t , $f(\mathbf{u})$ describes the growth and decline in concentration over time and $\mathbf{D} \nabla^2 \mathbf{u}$ describes the movement in space. Equation (2.4) may be used to model predator–prey interactions amongst species by defining the interaction as reactive (growth) and diffusive (movement) components. Models by Murray in 1976 [28] and Zheng in 1986 [47] demonstrate the efficacy of the reaction–diffusion equation to model the spatial dependence in predator–prey interactions.

After several decades of mathematically modelling predator–prey interactions, the need arose for a new type of model, specifically one that could model a system with many complexities without becoming as computationally expensive as spatially modelled differential equations [17].

2.2 Cellular automata

Apart from differential equations, one can model predator–prey interaction by means of explicit simulation. With the amount of computational power currently available, models such as CAs are becoming more popular [18].

2.2.1 Definition

A CA consists of a n -dimensional grid of cells, each one in a finite number of states. Each cell has a defined neighbourhood of bordering cells. Two popular neighbourhood definitions are the Von Neumann type and the Moore type [39], as illustrated in Figure 2.1.

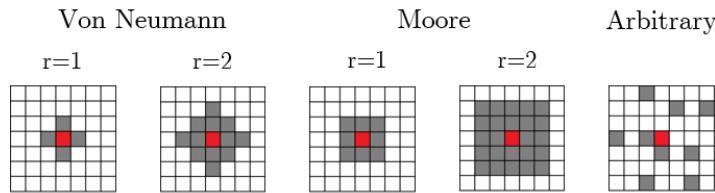


Figure 2.1: *Various neighbourhoods for a 2D cellular automaton, including radii greater than 1. The red cell indicates the current cell while the grey cells make up its neighbourhood.*

A CA is initialised by setting each cell to a starting state. Time then progresses in discrete steps, with every cell updating its state simultaneously during each time step according to a function or rule relating to the cell’s previous state and that of its defined neighbours. Other than initialisation, no other input is required from the user, making it a true automaton or “zero-player game”. It is important to note that, for an automaton to be considered a true CA, two rules must apply [35]:

1. All cells update simultaneously, not sequentially.
2. A rule applied to a cell only updates that specific cell.

2.2.2 A brief history

It is widely accepted that the first person to introduce the idea of the classical CA was Hungarian-American mathematician John von Neumann [45]. He described this “cellular space” as a two-dimensional infinite array of uniform cells, each connected to its northern, eastern, southern and western (or orthogonal) neighbours, known today as the Von Neumann neighbourhood set [31]. He introduced the CA as a means to model self-reproducing biological systems in 1966 [39], after already having expanded on the theory behind various automata in 1951 [38].

In 1970, British mathematician John Conway experimented with a simple two state 2D cellular automaton, applying sets of very complex rules using a cell’s Moore neighbourhood [14]. The resulting CA, the Game of Life, produced some noteworthy results, including a “machine” that

could generate other “machines” on a rectangular grid. Gardner [14] described it as the push CAs needed to be taken seriously as a viable simulation model for complex systems. CAs were again popularised in the 1980s by work done by fellow British mathematician, Stephen Wolfram [44, 45].

2.2.3 Applications in complex systems

Uses for CAs have been found in many fields, such as computer science, biology, ecology, geography and epidemiology. It was in 1984 that Toffoli [34] commented on the effectiveness of using CAs as an alternative to differential equations in physics. He noted that with increased computational power it was indeed a viable alternative, opening up the use of CAs for physical systems.

Geographical modelling and epidemiology

Applications of CAs often seen in literature are growth models, such as the urban expansion of cities, the spreading of forest fires and even communicable diseases through populations. Karafyllidis & Thanailakis [20] introduced a CA in 1997 that could model the spreading of forest fires in homogeneous and heterogeneous forests and accommodate changes in weather patterns and land topography, the first of its kind. In 1997, White *et al.* [42] used CAs to model the evolution of urban land usage over time for Cincinnati, Ohio. It illustrated growth based on location and resources available in an extensive 113-cell neighbourhood, achieving realistic results and thereby illustrating the effective use of CAs in planning. Fuentes & Kuperman [13] proposed using a CA to model the infectivity of a disease in a population, considering a virus-like disease that leaves no immunity after contracting it, and another disease that could leave an immunological response once contracted.

Computer science

During the 1980s, when CAs started gaining popularity again, Wolfram [43] proposed the use of one-dimensional automata in cryptography, exploiting the fact that an initial setup will always follow the same evolutionary steps. Chowdhury *et al.* [8] developed a CA-based approach to error correction coding that could fix single byte and double byte errors in transmitted code.

Biological and ecological models

Many applications of CAs to biological models exist in literature. Ermentrout & Edelstein-Keshet [11] compiled a host of applications based on previous works, including mimicking animal coat markings [46], host-parasitoid interactions [16] and branching of biological structures such as fungi [10]. The results obtained closely matched their mathematically modelled counterparts, further cementing the use of the CA in biological modelling.

CAs as a means to model predator-prey interactions have been prevalent since the early 1990s. Models have been built to observe the effect of harvesting on a predator-prey interaction model, aiming to determine the stable limits of harvesting different combinations of predators and prey with various harvesting techniques [29]. Models concerning more than two species have also been investigated, specifically where more than one predator species exists [17]. Apart from general predator-prey interaction models, some models have been designed around the interactions between specific species, such as the approach by Chen *et al.* [7] to model the interaction

between competitive growth in two underwater species in the Netherlands.

The instinctual responses of predator pursuit and prey evasion have been explored to an extent in the past. Boccara *et al.* [2] implemented a two stage automaton model in 1994 that separates the feeding and breeding cycle from the movement cycle. In 2006, Cattaneo *et al.* [4] improved on this model, modifying it and adding complexities, such as distinguishing between fed and unfed predators, and cells that were empty or cells that became empty due to death. They also introduced a rule that could facilitate natural death of prey to ensure the oscillatory nature of the system holds for any parameters not leading to extinction. This model is one of the most complete predator-prey interaction cellular automata in literature, and shall form the basis for the models developed in this study.

2.3 Spatial artificial life models

An artificial life model, or “animat”, is defined as any simulation or emulation of a life form or life forms where explicitly mapped individuals have the capacity to make their own decisions of how they interact with their environment. An individual is considered an “agent”, therefore animats are classified as agent-based simulations [18].

The advantage of using animats in modelling biological systems, such as predator-prey interactions, is the ability to visually represent the behaviour of individuals over time. One such observation is the tendency for prey to form a defensive spiral, as observed by Hawick *et al.* [18] in 2004. In their implementation of the predator-prey model, they found that, given an awareness and ability to “see” each other on the grid, that a grouping of prey being pursued by predators would tend to curl around at the edges. This resulted in the formation of the so-called defensive spiral in military terms. Another investigation led by Scogings & Hawick [32] modelled the effect of altruism¹ in a predator-prey interaction model. They found that new patterns emerged when either or both species displayed some level of altruism, especially when considering the stability of the model, as individuals could keep each other alive by feeding otherwise starved individuals.

Artificial life models occur frequently outside of academia and have applications that extend beyond that of modelling real-world interactions. In 2007, GSC Game World released S.T.A.L.K.E.R.: Shadow of Chernobyl, a first-person shooter survival horror video game. What set the game apart from others was its dynamic artificial intelligence model known as A-Life, where the game world was not scripted by the presence of the player. Every person, animal and monster has the ability to interact with the world and each other with its own life cycle and could do so autonomously. Predators and monsters would actively hunt prey if it was hungry and beings being preyed upon could try and evade capture [5]. The player character was effectively just another agent in the world with its life cycle determined by the user.

¹In animal behaviour, altruism is defined as acting in such a way that may be to the disadvantage of an individual, but benefits others of its kind. A warning cry is an example of altruistic behaviour in social animals.

CHAPTER 3

Methodology

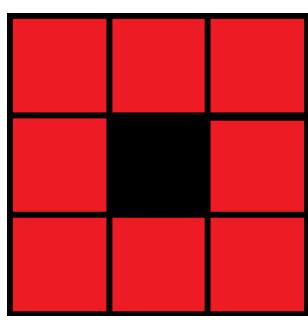
In this chapter, the logic behind the CA and how predator-prey interactions may be modelled with it will be discussed in §3.1 and §3.2. Three separate models of increasing complexity will then be presented along with the respective rules and assumptions made. The simplest model, where only feeding and reproduction occurs, is described in §3.3. The predator pursuit and prey evasion models are described in §3.4 and §3.5.

3.1 Cellular automaton

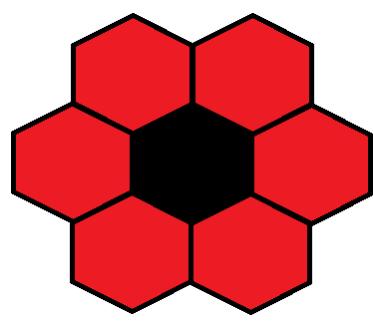
A CA, as defined in §2.2.1, consists of a grid of cells, with each cell in one of a finite number of states. In each discrete time step, the state of every cell is updated according to a set of rules applied to the cell and its defined neighbours. Once initialised, a CA runs without requiring user intervention. Two of the most important considerations when implementing a CA is its geometry and rules.

3.1.1 Geometry

Theoretically, the cells that form a two-dimensional CA can be of any shape, although square and hexagonal cells are the most common. In effect, the only difference the shape of a cell makes is the layout of its neighbourhood, as illustrated in Figures 3.1(a) and 3.1(b).



(a)



(b)

Figure 3.1: (a) A square cell in black with its eight cell Moore neighbourhood and (b) a hexagonal cell with its six cell Moore neighbourhood.

In physical systems, the two-dimensional array often represents a spatial domain. The array

can be any shape, with rectangular shapes being the most common. The boundaries depend on the geometry of the CA. Geometries for a CA may broadly be separated into two categories: toroidal and non-toroidal.

Toroidal

Toroidal geometry assumes that the top and bottom, left and right boundaries are connected, thereby forming a torus¹, and can be seen as periodic. When considering a rectangular plane, anything that crosses an edge immediately emerges on the opposite side of the plane, continuing in its original direction of travel. Many CAs, including the Game of Life, have toroidal geometry and may be considered realistic in physical systems that are spherical in shape.

Non-toroidal

Non-toroidal geometry implies that the boundary is fixed. A reflective boundary acts as a wall that cannot be crossed, and is used when modelling enclosed spaces such as bacteria in a petri dish. A dispersive boundary is not a physical boundary and allows movement across the edges. A model that considers a smaller portion of a large system, for example a square kilometre within a large wildlife preserve, may use a dispersive boundary.

3.1.2 Rules

The rules defined are what allow a CA to run autonomously. Rules may be deterministic, probabilistic or a combination of the two. To illustrate how rules work, consider a configuration in the Game of Life as illustrated in Figure 3.2. Rule 1 states that any live cell will remain alive if two or three of its neighbours in the Moore neighbourhood are alive [14]. By this rule, the cell marked 1 will stay alive in the next step. Cell 2, if we assume its unseen neighbours are all dead, would die as only cell 1 is alive in its neighbourhood. Rule 2 states that every dead cell with exactly three live neighbours will become alive. Cell 3, which is dead in the time step considered, has exactly three live neighbours and will be alive in the following step.

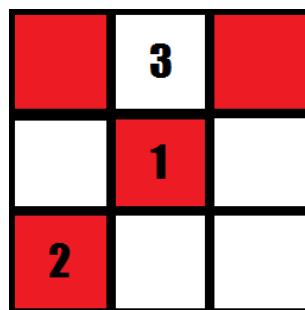


Figure 3.2: Nine cells illustrating a configuration at a certain time step in the Game of Life. Red represents alive cells and white represents dead cells.

The rules for the Game of Life are deterministic in nature, with no randomness included. Every run of the model with the same initial setup will therefore have identical progression. Again consider rule 1, but modify it such that a live cell with two live neighbours has a 40% chance of staying alive and an 80% chance of staying alive if it has three live neighbours. This modified

¹A torus is a surface or solid formed by rotating a closed curve, like a circle, about a line which lies in the same plane but does not intersect it, i.e. a doughnut.

rule is probabilistic, introducing randomness that leads to a unique progression for every run.

3.2 Predator-prey interactions

Consider a habitat of finite size containing two species, a and b , with b preying on a . Assume the energy source for a is abundant and b relies solely on a as a source of energy. Groups of a would naturally try and reproduce and evade predation, while b will actively prey on a in order to reproduce. It is logical to assume that an individual of b has to be physically close to individuals of a in order to catch and eat them, and that reproduction can only occur in the vicinity of any individuals. Such a scenario may be modelled with a CA as follows:

- Set grid \mathcal{Q} as a two-dimensional array with $m \times n$ cells, m and n being non-negative integers.
- Define the boundary conditions as in §3.1.1.
- Let each cell $s \in \mathcal{Q}$ be in one of three states: empty (0), prey (a) or predator (b). Let the states be a set $\mathcal{L} = \{0, a, b\}$.
- Create a set of rules, \mathcal{T} , that dictates the behaviour of each cell.

This model forms the basis of most two-species predator-prey CAs in literature, with most focusing on variations in the set of rules, \mathcal{T} .

3.3 The simple model

The first model developed in this study has the simplest set of rules, only running through a single step per iteration and applying all rules that are applicable to the specific cell simultaneously. The model is still comparable in its complexity with, and is based on a combination of models by Chen *et al.* [7], Chen & Mynett [6], and Hawick & Scogings [17] in 2010.

3.3.1 Assumptions and geometry

The relative simplicity of the model requires many assumptions regarding the environment and behaviour of the species. All members of the respective populations/species are homogeneous, meaning that there is no distinction between two members of the same species. Prey can only die by predation and is not sensitive to overpopulation. This will result in one hundred percent prey saturation of the array in the absence of predators. Predator death is a random process, independent of hunger or age, and every member has equal probability of dying each time step. The prey is the sole source of energy for the predators, and the food source for the prey is absolutely abundant. It is also assumed that the environment is homogeneous, i.e. there are no obstacles and every cell is equally accessible. Individuals do not have the ability to survey their surroundings and have no explicit instructions to move. The model is constructed in such a way to simulate movement by breeding in more desirable spaces. Every time step executes the same set of rules, ignoring any time-related cycles such as day/night or seasonal cycles.

The geometry of the CA is as follows:

- The cells are square/rectangular in shape.
- The boundary is reflective, implying a closed system.
- The neighbourhood is Moore-type of radius one, giving each cell eight neighbours, unless it is on the edge where it has five neighbours and in the corners where it has three.

3.3.2 Rules

The rules of the model execute in a single step, applying every rule applicable to a cell's state once before moving to the next cell. The full set of rules is set out in Algorithm 3.1.

Algorithm 3.1 Simple predator-prey rules

```

1: procedure APPLYRULES
2:    $d_p$  is prey death rate
3:    $b_p$  is prey birth rate
4:    $d_h$  is predator death rate
5:    $b_h$  is predator birth rate
6:   for all cells do
7:      $x \leftarrow$  next cell
8:     Evaluate state of  $x$ 
9:     if  $x$  is prey then
10:       $r \sim U(0, 1)$ 
11:      Evaluate Moore neighbourhood  $\mathcal{M}$  of  $x$ 
12:       $nPred =$  number of predators in  $\mathcal{M}$ 
13:      if  $r < (1 - d_p)^{nPred}$  then
14:        Hunt failed/no predators, cell stays prey
15:      else
16:         $r_h \sim U(0, 1)$ 
17:        if  $r_h < b_h$  then
18:          Cell becomes predator by breeding
19:        else if  $x$  is predator then
20:           $r \sim U(0, 1)$ 
21:          if  $r < d_h$  then
22:            Cell becomes empty due to predator death
23:          else
24:            Cell stays predator
25:        else if  $x$  is empty then
26:          Evaluate Moore neighbourhood  $\mathcal{M}$  of  $x$ 
27:           $nPred =$  number of predators in  $\mathcal{M}$ 
28:           $nPrey =$  number of prey in  $\mathcal{M}$ 
29:          if  $nPrey = 0$  or  $nPred > 0$  then
30:            Cell remains empty
31:          else
32:             $r \sim U(0, 1)$ 
33:            if  $r < (1 - b_p)^{nPrey}$  then
34:              Cell becomes prey by breeding

```

As illustration, consider a given cell, x . If the state of x is prey, its state will remain prey with probability $(1 - d_p)^{nPred}$, else the prey dies and the state of x becomes predator with probability

b_h , or empty (0) with probability $1 - b_h$. If the state of x is predator, its state becomes empty (predator death) with probability d_h , else it remains a predator. If x is empty and there is either no prey or some predators in its Moore neighbourhood, then it remains empty. Else the state of x will become prey with probability $(1 - b_p)^{n_{\text{Prey}}}$.

The model simulates prey death by a function of predators in its neighbourhood and a parameter d_p that indicates how successful a predator is in catching its prey. The same function applies to prey breeding with the exception of it being a function of prey in its neighbourhood and a birth parameter, b_p . It also requires at least one other prey individual in its neighbourhood. The probability values generated by these functions are illustrated in Figure 3.3.

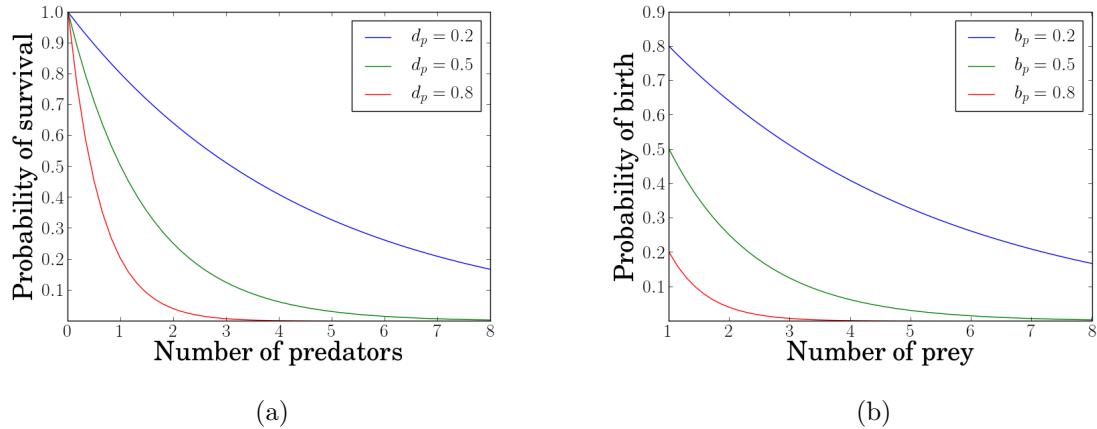


Figure 3.3: (a) The probability of prey surviving given a number of predators in its neighbourhood and (b) the prey being born given the number of prey in its neighbourhood.

3.3.3 Model validation

According to Verhulst [36], a prey population without predation should grow by the logistic function $\frac{dN}{dt} = rN(1 - \frac{N}{K})$. This behaviour may clearly be seen in Figure 3.4(a), a satisfactory result.

Considering the assumption that the predator's sole source of energy is the prey, a model containing only a predator population should immediately start declining until it reaches extinction. The shape of the curve is irrelevant, except that it must be strictly decreasing at all times. From Figure 3.4(b) it is clear that this is indeed the case and the decline is exponential.

In the case where there are populations of both predators and prey, stability with periodic oscillations is expected as illustrated by the Lotka–Volterra model. The population dynamics and phase graphs in Figure 3.5 shows that, while there are oscillations, they are not as periodic as the Lotka–Volterra model as the CA is stochastic in nature and the Lotka–Volterra model is deterministic. This is to be expected, as not only population size, but individual location is considered as well. The parameters for the test was $b_h = 0.2$, $b_p = 0.8$, $d_h = 0.2$ and $d_p = 0.8$ on a 50×50 array.

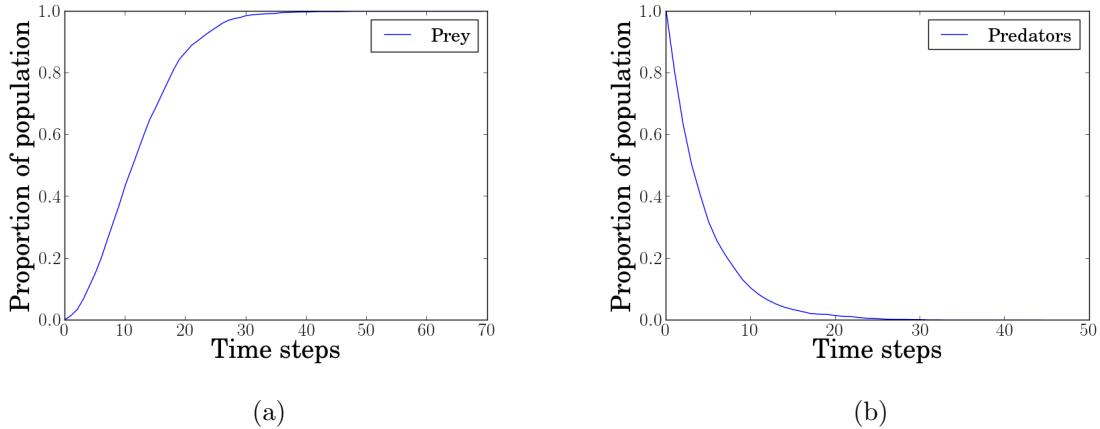


Figure 3.4: The simple model has (a) the prey grow logistically with $b_p = 0.2$ and (b) the predator population strictly decline with $d_h = 0.2$ in the absence of the other species.

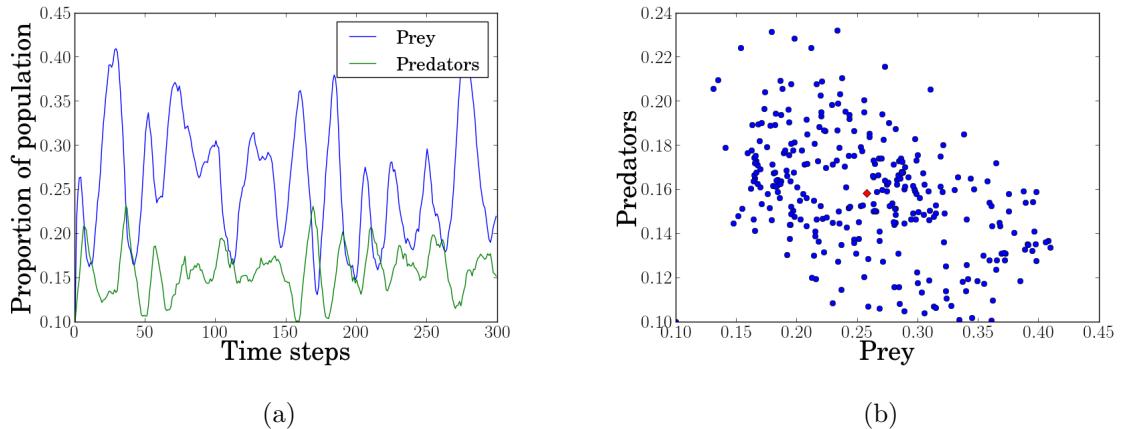


Figure 3.5: (a) The population dynamics and (b) the phase graph of the simple model showing some large yet periodic oscillations.

3.4 Predator pursuit and prey evasion

The predator-pursuit-prey-evasion (PPPE) model is based on the model of Cattaneo *et al.* [4], with the exception of the boundary conditions and predator actions when no prey is visible. It differs from the first model as it requires three sub-phases per iteration and simulates movement explicitly. Most assumptions made for the simple model are still applicable in the PPPE model.

3.4.1 Assumptions and geometry

As in the simple model, the environment as well as the members of the respective populations are homogeneous. Feeding occurs before reproduction, which itself precedes movement. Only predators that have eaten may be considered for reproduction, assuming that reproduction requires a certain amount of energy. Predators and prey both have the same field of vision and would always want to move to the most desirable cell in its neighbourhood. A prey individual with no intent to move will remain stationary, as its food source is abundant, but a predator with no prey in sight will embark on a random walk looking for food. All other assumptions

made for the simple model also apply to the PPPE model.

The geometry of the CA is as follows:

- The cells are square/rectangular in shape.
- The boundary is reflective, implying a closed system.
- The neighbourhood is Von Neumann-type of radius one, giving each cell four neighbours, unless it is on the edge where it has three neighbours and in the corners where they have two.
- The field of vision is in the Moore neighbourhood with variable radius r_v .

Individual movement is based on an intent system. Predators prefer moving towards areas of high prey density, and prey want to move to areas of low predator density. Cattaneo *et al.* [4] proposed dividing the extended Moore neighbourhood into four quadrants: north, east, south and west, as illustrated in Figure 3.6, forming four cones of vision. A mapping $T : \mathcal{L} \times \mathcal{Q} \mapsto \{0, 1\}$ is defined where

$$T(v, x, y) = \begin{cases} 1 & \text{if cell } (x, y) \text{ is } v \in \mathcal{L} \\ 0 & \text{otherwise.} \end{cases}$$

For the Moore neighbourhood of radius r , the following quantities may be assigned to any cell (x, y) and each state $v \in \mathcal{L}$:

$$n_N^{(r_v)}(v; x, y) = \sum_{i=1}^r \sum_{j=-i}^i T(v, x + j, y + i), \quad (3.1)$$

$$n_E^{(r_v)}(v; x, y) = \sum_{i=1}^r \sum_{j=-i}^i T(v, x + i, y + j), \quad (3.2)$$

$$n_S^{(r_v)}(v; x, y) = \sum_{i=1}^r \sum_{j=-i}^i T(v, x + j, y - i) \text{ and} \quad (3.3)$$

$$n_W^{(r_v)}(v; x, y) = \sum_{i=1}^r \sum_{j=-i}^i T(v, x - i, y + j) \quad (3.4)$$

where $n_D^{(r_v)}(v; x, y)$ denotes the number of individuals of type v (predator or prey) in the north (N), east (E), south (S) and west (W) quadrants of the Moore neighbourhood of radius r_v , respectively, for a cell at position (x, y) . These values are then used to determine the movement intent of the individual in cell (x, y) . If more than one individual wishes to move to a specific empty cell, one of them is randomly chosen to move and the others remain stationary. An individual can only move to an empty cell, and if its first choice is not available, it remains stationary. It should be noted that all cells on the edge of two cones of vision count towards both cones.

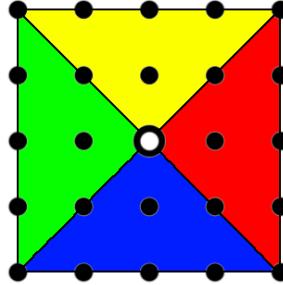


Figure 3.6: *The four quadrant view cone of the Moore neighbourhood with radius 2. Yellow is north, red is east, blue is south and green is west.*

3.4.2 Rules

In the PPPE model, every iteration is subdivided into three phases, namely, feeding, reproduction and movement. Each phase is executed one after the other, utilising the configurations generated by the phase before it, i.e. the reproduction phase applies its rules to the array generated by the feeding phase and the movement phase applies its rules to the array generated by the reproduction phase.

Feeding

For the feeding phase of the algorithm, it is necessary to define a new set of states \mathcal{L}_2 . Let 0_a denote a standard empty cell, and 0_b a cell that is empty because a kill occurred there. Also, let b_0 denote a predator that does not eat in the respective phase and let b_1 denote a predator that has eaten. Prey is still denoted by a . The new set of states is $\mathcal{L}_2 = \{0_a, 0_b, a, b_0, b_1\}$.

Algorithm 3.2 is similar to Algorithm 3.1, with the exception of predator deaths and reproduction in empty cells. The manner in which prey cells are handled is virtually identical.

Algorithm 3.2 Feeding phase of the PPPE

```

1: procedure FEEDINGPHASE
2:    $d_p$  is prey death rate
3:   for all cells do
4:      $x \leftarrow$  next cell
5:     Evaluate state of  $x$ 
6:     if  $x$  is prey ( $a$ ) then
7:        $r \sim U(0, 1)$ 
8:       Evaluate Von Neumann neighbourhood  $\mathcal{V}$  of  $x$ 
9:        $nPred =$  number of predators in  $\mathcal{V}$ 
10:      if  $r < (1 - d_p)^{nPred}$  then
11:        Hunt failed/no predators, cell stays prey ( $a$ )
12:      else
13:        Cell becomes empty due to kill ( $0_b$ )

```

```

14:   else if  $x$  is predator ( $b_0$ ) then
15:      $r \sim U(0, 1)$ 
16:     Evaluate Von Neumann neighbourhood  $\mathcal{V}$ 
17:      $nPred =$  number of predators in  $\mathcal{V}$ 
18:     if  $r < (1 - d_p)^{nPred}$  then
19:       The hunt fails, predator stays  $b_0$ 
20:     else
21:       The hunt succeeds, predator is now fed ( $b_1$ )

```

As illustration, consider a cell x in Algorithm 3.2. If the state of x is prey (a), the state of x remains prey with probability $(1 - d_p)^{nPred}$, else the prey dies and x becomes an empty cell due to a kill (0_b). If the state of x is predator (b_0), the predator fails the hunt with probability $(1 - d_p)^{nPred}$ and the state of x remains (b_0). Else the hunt succeeds and the state of x becomes a fed predator (b_1).

Reproduction

After the feeding phase, both species can reproduce and predators can die. Unlike in the simple model, predators are required to have eaten before they are able to reproduce, adding a degree of realism to the simulation. The outline of the reproduction phase may be seen in Algorithm 3.3, where the new states in the resulting array configuration, obtained by applying Algorithm 3.2, determine the reproductive behaviour of the respective species. Reproduction probabilities are similar to those in the simple model.

Algorithm 3.3 Reproduction phase of the PPPE

```

1: procedure REPRODUCTIONPHASE
2:    $b_p$  is prey birth rate
3:    $d_h$  is predator death rate
4:    $b_h$  is predator birth rate
5:   for all cells do
6:      $x \leftarrow$  next cell
7:     Evaluate state of  $x$ 
8:     if  $x$  is prey ( $a$ ) then
9:        $x$  remains prey
10:      else if  $x$  is predator ( $b_0$  or  $b_1$ ) then
11:         $r \sim U(0, 1)$ 
12:        if  $r < d_h$  then
13:          The predator dies, cell is now empty ( $0_a$ )
14:        else
15:          The predator lives and becomes  $b_0$ 
16:      else if  $x$  is an empty cell ( $0_a$ ) then
17:        Evaluate Von Neumann neighbourhood  $\mathcal{V}$  of  $x$ 
18:         $nPred =$  number of predators in  $\mathcal{V}$ 
19:         $nPrey =$  number of prey in  $\mathcal{V}$ 
20:        if  $nPred = 0$  or  $nPred > 0$  then
21:          Cell remains empty ( $0_a$ )
22:        else
23:           $r \sim U(0, 1)$ 
24:          if  $r < (1 - b_p)^{nPrey}$  then
25:            Cell becomes prey ( $a$ ) by breeding

```

```

26:   else if  $x$  is a cell empty due to a kill ( $0_b$ ) then
27:     Evaluate Von Neumann neighbourhood  $\mathcal{V}$  of  $x$ 
28:      $nFedPred$  = number of fed predators ( $b_1$ ) in  $\mathcal{V}$ 
29:      $r \sim U(0, 1)$ 
30:     if  $r < (1 - b_h)^{nFedPred}$  then
31:       No reproduction occurs, cell becomes standard empty ( $0_a$ )
32:     else
33:       Cell becomes predator ( $b_0$ )

```

As illustration, consider a cell x in Algorithm 3.3. If the state of x is prey (a), it remains prey. If the state of x is any predator (b_0 or b_1), the cell becomes empty (predator death) with probability d_h , else the state of x remains a predator (b_0). If the state of x is empty, but not due to a kill in the feeding phase, its Von Neumann neighbourhood is evaluated. If there is either no prey or some predators, x remains empty, else the state of x becomes prey with probability $(1 - b_p)^{nPrey}$. If the state of x is empty due to a kill in the feeding phase, it will remain empty with probability $(1 - b_h)^{nFedPred}$, else the state of x becomes predator (b_0). All cells with a predator state (b_0 and b_1) have their states changed to unfed predator (b_0) during the predator death phase.

Movement

As stipulated in §3.4.1, movement is based on an intent system that has individuals move to cells of greatest desirability. The reflective boundary prohibits individuals from moving across borders, however, there can still be intent to move through the boundary, in which case the individual remains stationary. The outline of the movement phase is given in Algorithm 3.4. The array configuration obtained by applying Algorithm 3.3 is used as an input for Algorithm 3.4.

Algorithm 3.4 Movement phase of the PPPE

```

1: procedure MOVEMENTPHASE( $r_v$ )
2:   for all cells do
3:      $x \leftarrow$  next cell
4:     Evaluate state of  $x$ 
5:     if  $x$  is prey then
6:       Set  $v$  to predator ( $b_0$ )
7:       Calculate  $n_N^{(r_v)}$  by equation (3.1)
8:       Calculate  $n_E^{(r_v)}$  by equation (3.2)
9:       Calculate  $n_S^{(r_v)}$  by equation (3.3)
10:      Calculate  $n_W^{(r_v)}$  by equation (3.4)
11:      Choose the minimum of  $[n_N^{(r_v)}, n_E^{(r_v)}, n_S^{(r_v)}, n_W^{(r_v)}]$  as intent
12:      if intent is null then
13:         $x$  has no intent, remains stationary
14:      if intent cell is empty then
15:        Add intent to competition list for  $x$ 
16:      else
17:         $x$  remains stationary
18:      else if  $x$  is predator then
19:        Set  $v$  to prey ( $a$ )

```

```

20:      Calculate  $n_N^{(r_v)}$  by equation (3.1)
21:      Calculate  $n_E^{(r_v)}$  by equation (3.2)
22:      Calculate  $n_S^{(r_v)}$  by equation (3.3)
23:      Calculate  $n_W^{(r_v)}$  by equation (3.4)
24:      Choose the maximum of  $[n_N^{(r_v)}, n_E^{(r_v)}, n_S^{(r_v)}, n_W^{(r_v)}]$  as intent
25:      if intent is null then
26:           $x$  has no intent, choose random intent
27:      if intent cell is empty then
28:          Add intent to competition list for  $x$ 
29:      else
30:           $x$  remains stationary
31:      for all cells in competition list do
32:           $x \leftarrow$  next cell
33:          if  $x$  has intent from more than one cell then
34:              Randomly select cell  $y$  in competition list
35:          Move chosen individual from cell  $y$  to cell  $x$ 

```

As illustration, consider a cell x in Algorithm 3.4. If the state of x is prey, the four quadrants (as illustrated in Figure 3.6) are evaluated for predators and an intent to move to the quadrant with the minimum number of predators is created. If there is no intent (no predators can be seen), the individual remains stationary, else the intent for x is added to a global competition list. If the state of x is predator, the four quadrants are evaluated for prey and an intent to move to the quadrant containing the largest number of prey is created. If there is no intent, a random intent is generated (predator starts a random walk) and added to a global competition list, else the intent for x is added to a global competition list. After every cell has been evaluated, the competition list is evaluated. If cell y in the competition list is only competed for by one other individual in cell, z , the individual moves from cell z to cell y . If more than one individual is competing to move to cell y , one is randomly chosen to move to y .

3.4.3 Model validation

The PPPE model should behave in a similar manner to the simple model when populated only with a single species. The logistic growth of the prey should be faster for the PPPE model, however, as only the smaller Von Neumann neighbourhood is used. Predator death should be similar to that seen in Figure 3.4(b). In Figure 3.7(b), a slower logistic growth is evident when compared to Figure 3.4a of the simple model, with marginally more time steps required to fill the array. As expected, given the same parameter values, predator extinction is identical in both models.

When running the PPPE model with the same parameter values as in §3.3.3, it is clear that the proportion of predators in the total population is larger and the prey population more stable when compared to the simple model, as illustrated in Figure 3.8(a). After the initial population spikes, the system oscillates relatively periodically, akin to oscillations found in more complex PDE and ODE models.

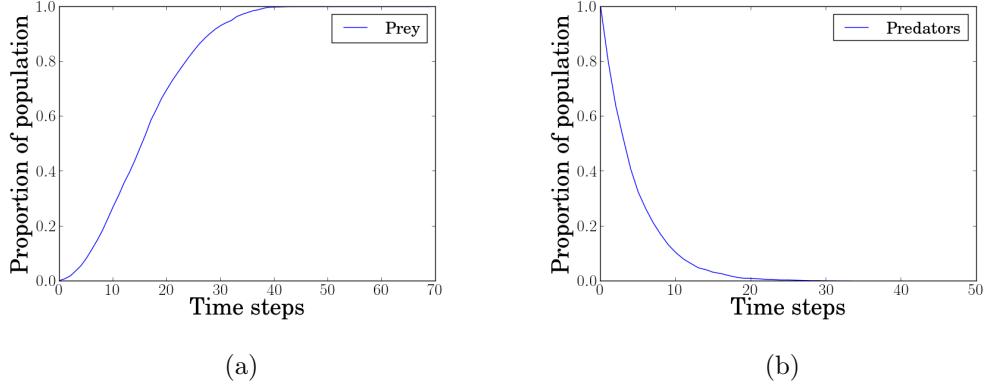


Figure 3.7: The PPPE model has (a) the prey grow logistically with $b_p = 0.2$ and (b) the predator population strictly decline with $d_p = 0.2$ in the absence of the other species.

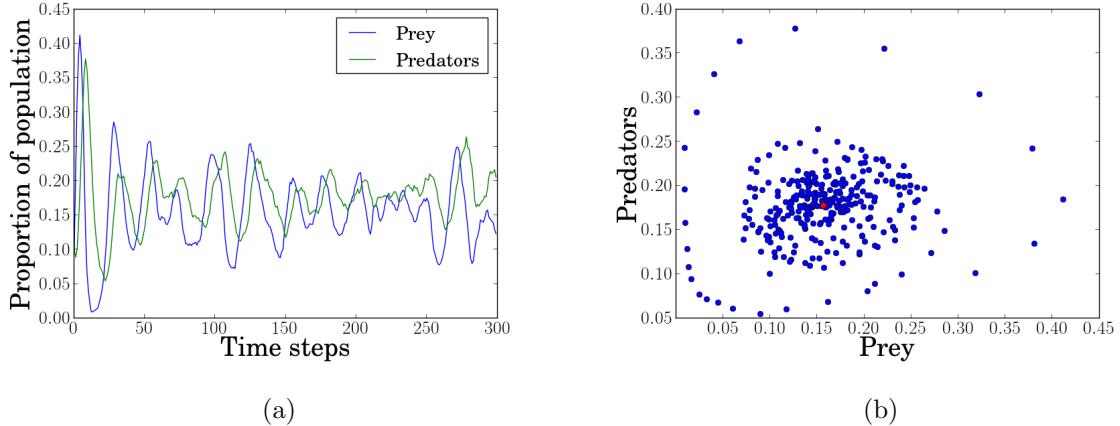


Figure 3.8: (a) The population dynamics and (b) the phase graph of the PPPE model showing some periodic oscillations. Both populations are roughly equal in size.

3.5 Density sensitivity and age adjusted mortality

The PPPE model, although an improvement over the simple model, contains some unrealistic assumptions with respect to unbounded prey growth and the random process governing predator death. The assumption that the prey population would grow until the entire grid is saturated counters the fact that any habitat, even with an abundant supply of energy, has some form of carrying capacity in terms of how densely the space can be packed. To remedy this, a method to simulate death by overpopulation for each member of the prey population is introduced, calculated using the density of the Moore neighbourhood of radius r_d . The assumption that the predator death probability is constant regardless of age is also adjusted to being age dependent, adding more realism to their behaviour.

3.5.1 Assumptions and geometry

Unlike the PPPE model, the improved model with density sensitivity and age adjusted mortality (DSAM model), has a heterogeneous predator population, with age being an added attribute. In addition, the prey population now has a natural death process due to overpopulation, implying that the population would not grow to completely saturate the space, but grow to some critical

level and oscillate around that level. Predator death is a function of age, allowing for a more realistic lifespan. All other assumptions made for the PPPE and simple model still apply and the geometry is identical to that of the PPPE model.

3.5.2 Rules

Two approaches are implemented for density sensitivity and one for age adjusted mortality. For density sensitivity, two functions presented by Cattaneo *et al.* [4] are included in the feeding phase of the PPPE model. The base function,

$$p(b_p, r_d) = n_d^{r_d(x,y)} \cdot f(b_p)/(2r_d + 1)^2 \quad (3.5)$$

assigns the probability of death for the prey individual in cell (x, y) due to the number of prey in its Moore neighbourhood ($n_d^r(x, y)$) of radius r_d per unit area. The function $f(b_p)$, with b_p the prey birth parameter, is a mapping on the interval $[0, 1]$ with two candidate functions:

$$f(b_p) = 1 - \cos\left(\frac{\pi}{2} \cdot b_p\right), \quad (3.6)$$

$$f(b_p) = 1 - e^{-e \cdot b_p}. \quad (3.7)$$

The function $p(b_p, r_d)$, plotted for various values of b_p and a radius of 2 is given in Figures 3.9a and 3.9b.

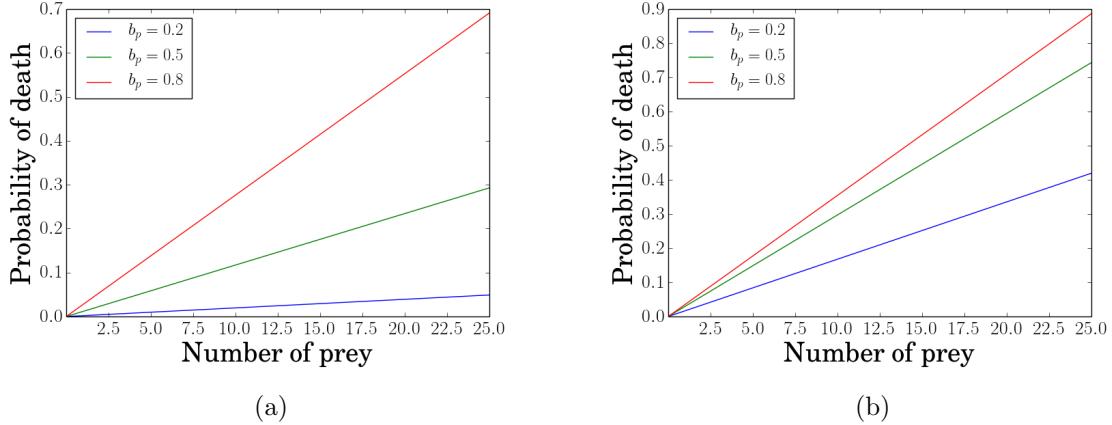


Figure 3.9: The probability of death due to density sensitivity generated by Equation (3.5) using (a) equation (3.6) and (b) equation (3.7) as $f(b_p)$.

Implementing density sensitivity for prey, consider again Algorithm 3.2 where x is evaluated as prey. If at any point prey survives, either due to there being no predators nearby or a nearby predator failing the hunt, prey can die due to overpopulation as outlined in Algorithm 3.5.

An age adjusted mortality rate for predators was implemented in an intuitive way. If it is assumed that a predator individual has a greater probability of death as it ages, it may be argued that the probability of death would increase in a nearly linear fashion. The linear function,

$$d_h(i) = \frac{i}{\alpha}, \quad (3.8)$$

is considered where $1/\alpha$ is the rate at which the probability of death increases with age i . One limitation with this approach is that the maximum age any predator can reach is α , as the probability of death would then be 1. Realistically, it may be argued that most, if not all,

species have some maximum age, where the probability of any individual reaching or surpassing this age is negligible, e.g. very few humans surpass the age of 100.

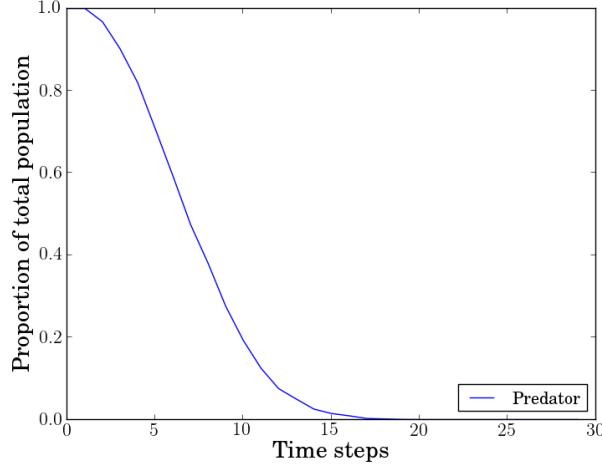


Figure 3.10: Setting the predator death probability to $\frac{x}{30}$. Most individuals die by “age” 7.

Incorporating the new predator death parameter, a simple replacement of d_h in line 12 of Algorithm 3.3 (reproduction phase) with equation (3.8) is required. If an individual does not die in a time step, it will age by 1 time step, i.e. i is incremented by 1.

Algorithm 3.5 Prey density

```

1: procedure FEEDINGPHASE
2:    $d_p$  is prey death rate
3:    $b_p$  is prey birth rate
4:   if  $x$  is prey ( $a$ ) then
5:      $r \sim U(0, 1)$ 
6:     Evaluate Von Neumann neighbourhood  $\mathcal{V}$  of  $x$ 
7:      $nPred =$  number of predators in  $\mathcal{V}$ 
8:     if  $nPred = 0$  then
9:       Evaluate Moore neighbourhood of radius  $r_d$ ,  $\mathcal{M}$ , of  $x$ 
10:       $np =$  number of prey in  $\mathcal{M}$ 
11:       $nh =$  number of predators in  $\mathcal{M}$ 
12:      if  $nh = 0$  then
13:         $prob = np \cdot f(b_p)/(2r_d + 1)^2$ 
14:        if  $r < prob$  then
15:           $x$  becomes  $0_a$  by prey death due to overpopulation
16:        else
17:           $x$  remains prey

```

```

18:   else
19:     if  $r < (1 - d_p)^{nPred}$  then
20:       Evaluate Moore neighbourhood of radius  $r_d$ ,  $\mathcal{M}$ , of  $x$ 
21:        $np$  = number of prey in  $\mathcal{M}$ 
22:        $nh$  = number of predators in  $\mathcal{M}$ 
23:        $prob = np \cdot f(b_p) / (2r_d + 1)^2$ 
24:       if  $r < prob$  then
25:          $x$  becomes  $0_a$  by prey death due to overpopulation
26:       else
27:          $x$  remains prey

```

As illustration, consider a cell x in Algorithm 3.5. If the state of x is prey (a), evaluate the Von Neumann neighbourhood for predators. If no predators are found, evaluate the Moore neighbourhood of radius r_d . If no predators are found, the state of x becomes empty (prey death) due to overpopulation with probability $p = np \cdot f(b_p) / (2r_d + 1)^2$, else it remains prey. If there were predators found in the Von Neumann neighbourhood, the hunt fails with probability $(1 - d_p)^{nPred}$ and the Moore neighbourhood of radius r_d is evaluated. Cell x will then be empty (prey death) by overpopulation with probability $p = np \cdot f(b_p) / (2r_d + 1)^2$, else it remains prey. If the hunt succeeded, the state of x becomes an empty cell where a kill occurred (0_b).

3.5.3 Model validation

From Figure 3.10, it may be seen that the predator population in the absence of food is strictly decreasing, as required. The prey population should still show logistic growth, but would stabilise at some point depending on the parameters b_p and r_d in equation (3.5), and then oscillate about this point as local prey densities fluctuate. Figures 3.11a and 3.11b illustrate this behaviour.

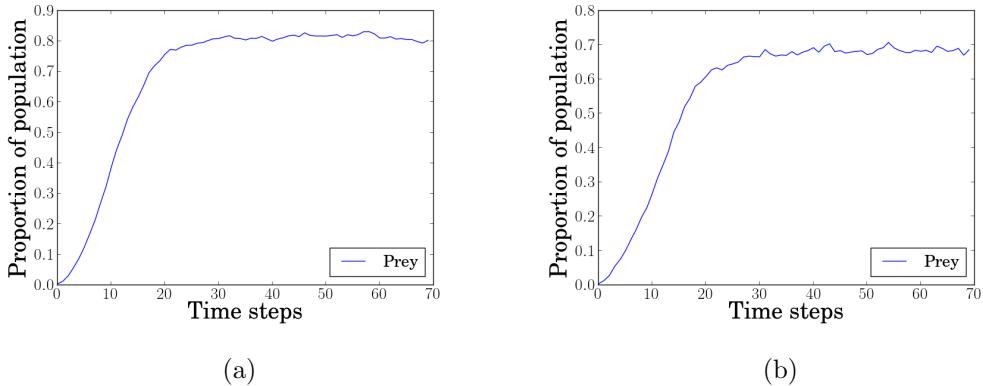


Figure 3.11: The prey population dynamics in the DSAM model with $b_p = 0.35$ and $r_d = 2$ for $f(b_p)$ set to (a) equation (3.6) and (b) equation (3.7).

Running the DSAM model with both species given the same parameters as in §3.3 and §3.4, setting r_d in equation (3.5) to 2 and α in equation (3.8) to 20 yields oscillations similar to those given by the Lotka–Volterra equations. In Figure 3.12, the results for $f(b_p)$ set to the cosine function in equation (3.6) is given.

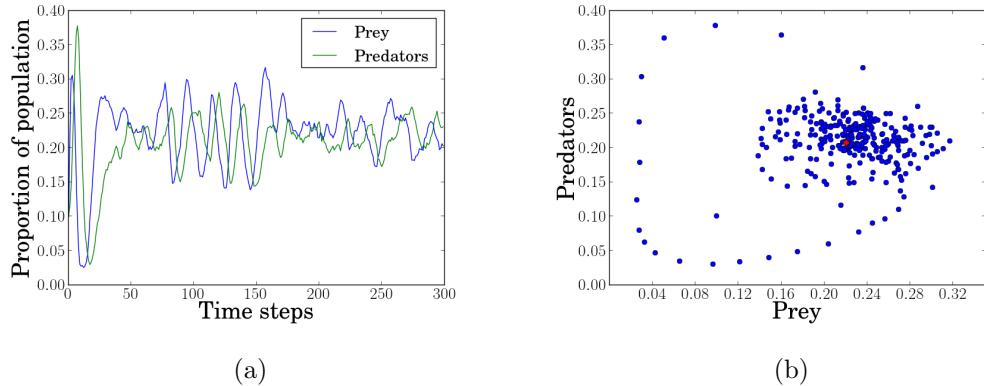


Figure 3.12: (a) The population dynamics and (b) the phase graph of the DSAM model showing some periodic oscillations. Both populations are roughly equal in size.

CHAPTER 4

Results

Available literature often lacks in-depth results and analysis of developed models. The analysis done on the three models presented in Chapter 3 is reported on in §4.2, with the fixed parameters and experiment configurations listed in §4.2.1. Results from parameter testing are given in §4.2.2, §4.2.3 and §4.2.4 for the simple, PPPE and DSAM models, respectively.

4.1 Model implementation

The models were implemented in PYTHON 3.3, and parameter testing was done on Dell Optiplex 9010 desktop computers with an Intel Core i7-3770 3.40 GHz processor and 4 GB of DDR3 RAM. The source code is available in Appendix B.

4.2 Parameter testing

An important part of the analysis of any simulation model is testing the model for various parameter values. Each of the three models developed has their own set of parameters, with more parameters added as complexity increases from the simple model to the DSAM model. How the parameters affect the models can be quantified by observing the population dynamics over a fixed time period. Observing whether or not two species can coexist given a set of parameters is arguably the most important result, as extinction of one or both species implies that the system is very unstable. Tables containing the numerical data for the graphs in this chapter are available in Appendix A.

4.2.1 Fixed parameters and configuration

For the purposes of experimentation with only the parameters affecting the rules of the respective model, it was necessary to implement a fixed configuration that would remove some of the randomness from the results. Differences in population dynamics should be solely as a result of the varying parameters in the rules and not the randomness of an initial configuration that could be biased towards one species or the other. Initial population distributions may be considered as a separate parameter to be analysed while keeping all other parameters constant. It was therefore decided to generate a fixed initial population for all experiments. The configuration used in the parameter tests are as follows:

- the array is a 50×50 cell square,

- the initial populations are 250 (10% of the array) for both species and are uniformly spread across the array,
- the CA will run for 300 iterations, and
- every set of parameters will be tested several times.

Sources in the literature do not specify how many times each simulation was run. Due to the condition that the initial population is identical for every experiment, a total of five runs per parameter set was considered to be sufficient as the results do not deviate significantly. The similarity of the outputs for five runs in all models may be given in Table 4.1. All models have a coefficient of variation (CV) of less than 0.1, indicating that the standard deviation between runs is small [30].

Run	1	2	3	4	5	Mean	Standard deviation	CV
Simple Model								
Prey	0.8470	0.8491	0.8561	0.8493	0.8542	0.8511	0.0034	0.0040
Predators	0.0010	0.0008	0.0008	0.0010	0.0009	0.0009	0.0000	0.0556
PPPE Model								
Prey	0.9752	0.9743	0.9730	0.9747	0.9749	0.9744	0.0008	0.0008
Predators	0.0008	0.0008	0.0010	0.0009	0.0008	0.0008	0.0000	0.0625
DSAM Model								
Prey	0.3422	0.3378	0.3374	0.3223	0.3361	0.3352	0.0067	0.0199
Predators	0.1067	0.1093	0.1083	0.1144	0.1110	0.1099	0.0027	0.0246

Table 4.1: *Deviation of the mean population sizes for the simple model over 5 runs.*

4.2.2 Simple model

The simple model, along with having the simplest set of rules, has the fewest number of parameters. The four parameters are

- b_p , the probability that a prey individual does not reproduce,
- d_p , the probability that a prey individual survives a predatory attack,
- b_h , the probability that a predator individual reproduces, and
- d_h , the probability that a predator individual dies.

The parameters may be divided into reproduction related (b_p and b_h) and mortality related (d_p and d_h) parameters. The parameter values [$b_p = 0.5, d_p = 0.5, b_h = 0.5, d_h = 0.5$] were set as the base case for all experiments. A total of 162 parameter combinations, using the value set [0.2, 0.35, 0.5, 0.65, 0.8], were analysed.

Effects of parameter variations on prey

The rules specific to prey individuals are dependent on the predators surrounding it. No new prey can be born with predators in its eight cell Moore neighbourhood, and prey can only die by predation. The prey population will indirectly be affected by b_h and d_h , as they determine the size and growth potential of the predators.

The proportion change in the mean prey population given a variation in a single parameter's value is illustrated in Figure 4.1. The population mean with respect to the base case, as stipulated in §4.2.2, is indicated by the purple marker. It is clear that the population mean did not change significantly if any one of the four parameters were changed from 0.5 to 0.2. All four population means lie within 3% of the base case, indicating that the prey population is not particularly sensitive to parameter variations below 0.5, except where $b_p = 0.35$, where the prey population induced predator extinction quickly and grew rapidly in their absence. Increasing b_p and b_h from 0.5 to 0.8 did have a significant impact on the prey population. Increasing d_p and d_h , the mortality parameters, showed very little change in the mean prey population, but increasing either b_p or b_h had a pronounced negative effect. Increasing b_p to 0.8, which decreases the probability that a prey individual will be born, gave an approximately 40% reduction in the prey population as their growth potential is severely limited.

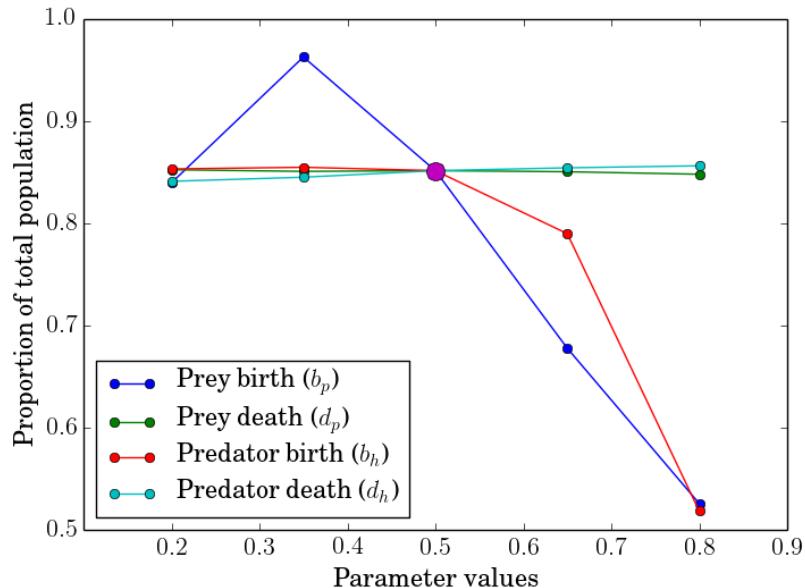


Figure 4.1: *The effect of single parameter variations on the mean prey population in the simple model.*

Unlike b_p , an increase in b_h will result in an increased probability that a predator will be born. Figure 4.1 illustrates that increasing b_h from 0.5 to 0.8 also lead to an approximately 40% reduction in the mean prey population. This indicated that the effects of reduced prey reproduction and increased predator reproduction had nearly equal effects on the mean prey population, with b_h having a greater effect for values above 0.65. Although the prey population fluctuated significantly with various parameter values, a lack of any pressure on the population, apart from predation, resulted in the population never reaching zero, i.e. going extinct.

Effects of parameter variations on predators

Rules describing the birth and death of predator individuals are not very dependent on the surrounding prey population, except that predators can only be born in cells where prey has died. The rule describing predator death is completely independent from all other birth and death processes in the CA, therefore the value for d_h effectively describes the average number

of predators that are removed from the system per time step, regardless of position.

The proportion change in the mean predator population given a variation in a single parameter's value is illustrated in Figure 4.2. The population mean with respect to the base case, as stipulated in §4.2.2, is indicated by the purple marker. The close proximity to 0 is due to the base case showing eventual extinction of the predator population. Reducing any of the parameter values, with the exception of b_p , and increasing any of the parameter values, with the exception of b_h , had a very small effect on the mean predator population and also resulted in extinction. As opposed to the prey population, reducing b_p had a positive effect on the predator population. The faster growing prey population allowed for more opportunities for predator births (as they are linked) and ultimately lead to a greater mean predator population. An increase in b_h had the most pronounced effect on the mean predator population. Since predators can only be born in cells which contained a prey individual in the previous time step, increasing the probability of predators being born will result in a larger proportion of predators in areas of high prey density (where kills occurred), leading to more prey potentially being killed and more predators being born, ultimately creating a sustainable cycle for the predator population.

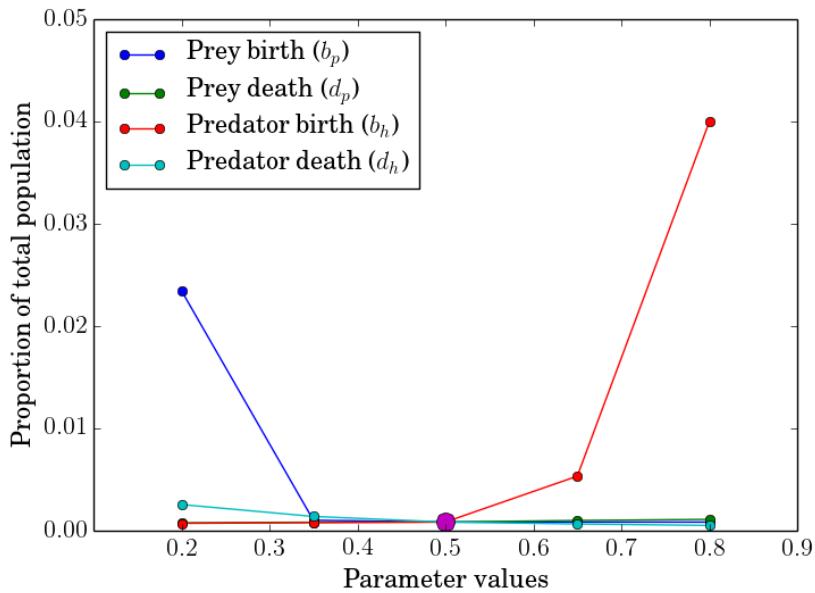


Figure 4.2: *The effect of single parameter variations on the mean predator population in the simple model.*

Unlike the prey population, there is a considerable amount of pressure on the predator population as they require kills to reproduce and their death process is completely random and independent from any other process. The increased pressure leads to populations that are at a greater risk of extinction. The number of parameter combinations tested per parameter value sums to 27. Of the 162 tests performed, 124 parameter combinations ultimately lead to predator extinction. The distribution of extinctions by parameter is illustrated in Table 4.2, with b_p and b_h again having the most significant effect. It may subsequently be concluded that the predator population is at much greater risk than the prey population.

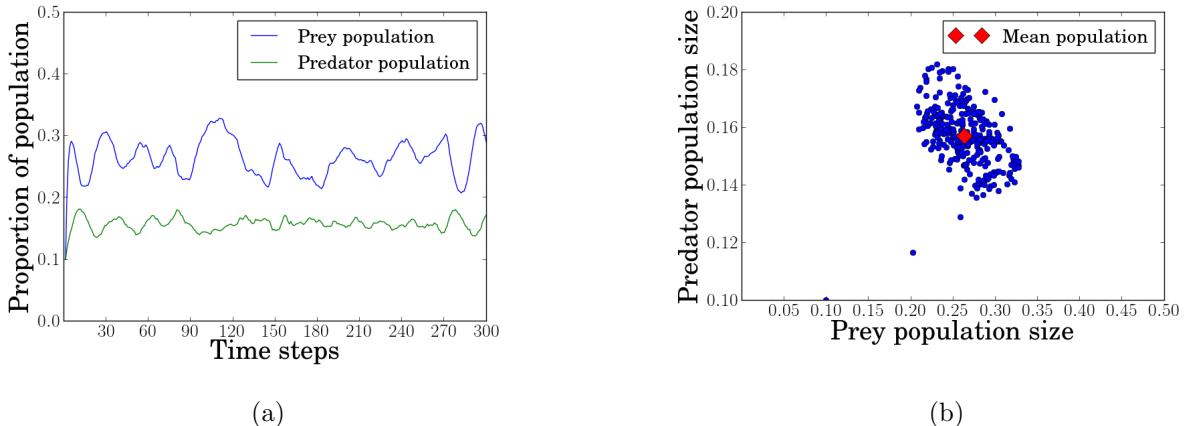
Parameter value	b_p	d_p	b_h	d_h
0.2	14	24	27	19
0.5	20	19	22	20
0.8	27	18	12	22

Table 4.2: *The number of predator extinctions per parameter value for the simple model.*

Review of the simple model

Observing the effects of various parameter values on the population dynamics yielded predictable results. From the results it may be concluded that the parameters related to the reproductive capabilities of the respective species have a greater effect on the populations than the death parameters. In the range for which parameters were tested, it may also be concluded that, for coexistence to occur, parameters favouring predators need to be used as they have a much greater risk of extinction.

If a set of parameter values is chosen that would most favour the predator population, the maximum mean predator population as well as the minimum mean prey population is found. The system is stable and the populations oscillate in a manner that is similar to the Lotka-Volterra equations, as illustrated in Figure 4.3(a). The oscillations are also relatively small, as the deviation from the population mean is small, as illustrated in Figure 4.3(b). The progression of the algorithm when rendered graphically is illustrated in Figure 4.4, indicating that prey individuals form clusters with predators attacking from the outside.

Figure 4.3: *Population dynamic (a) and phase (b) graphs for the simple model with parameters that favour predators.*

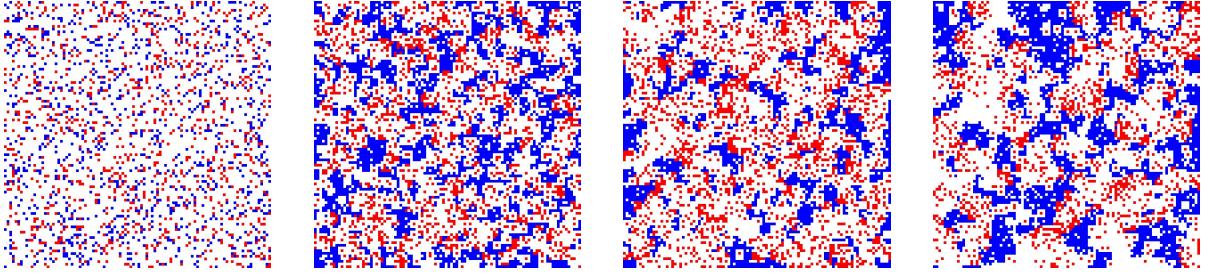


Figure 4.4: *Snapshots of the progression of the simple model for times $t = 1$, $t = 5$, $t = 10$ and $t = 20$ for a 100×100 array with parameters $[0.2, 0.8, 0.8, 0.2]$. Predators are shown in red and prey in blue.*

4.2.3 PPPE model

For the PPPE model, a radius parameter is added to the parameters of the simple model. The five parameters are

- r_v , the radius of the view cone as illustrated in Figure 3.6,
- b_p , the probability that a prey individual does not reproduce
- d_p , the probability that a prey individual survives a predatory attack,
- b_h , the probability that a predator individual reproduces, and
- d_h , the probability that a predator individual dies.

The parameters may be divided into spatial awareness, r_v , reproduction related, b_p and b_h , and mortality related, d_p and d_h . A total of 502 parameter combinations were used during testing, with the same value set for the probabilities (b_p, b_h, d_p, d_h) as for the simple model in §4.2.2 and [2, 3, 4, 5] for the radius, with the parameter values $[r_v = 3, b_p = 0.5, d_p = 0.5, b_h = 0.5, d_h = 0.5]$ set as the base case for all experiments.

Effects of parameter variations on prey

As with the simple model, the rules specific to prey are still dependent on the predators surrounding it, although the use of the smaller Von Neumann neighbourhood would have a direct impact on how significant the effects will be.

The proportion change in the mean predator population given a variation in a single parameter's value is illustrated in Figure 4.5. The population mean with respect to the base case, as stipulated in §4.2.3, is indicated by the purple marker. The model is more sensitive to an increase in b_p , d_p and b_h and more sensitive to a decrease in d_h . A decrease in the predator death rate d_h had a particularly significant effect on the system, adding a large amount of pressure on the prey population. In general, the mortality parameters had a more pronounced effect on the populations than the reproduction parameters.

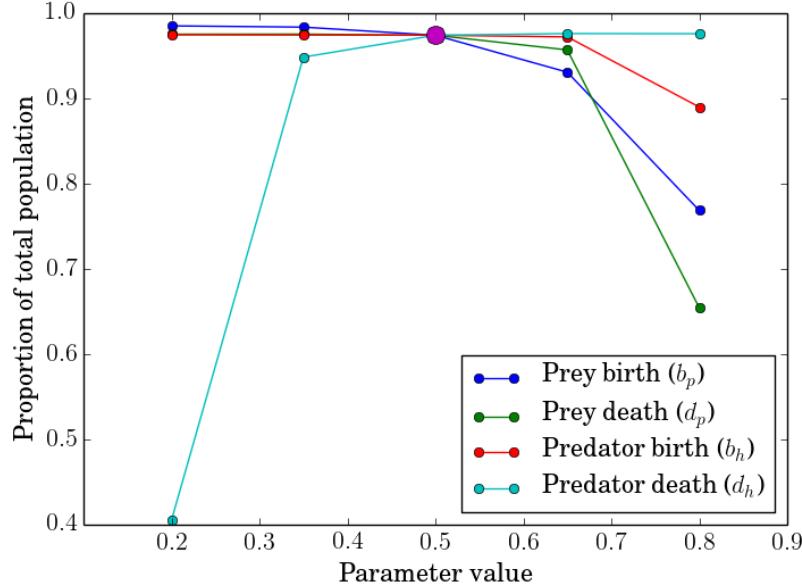


Figure 4.5: *The effect of single parameter variations on the prey population in the PPPE model.*

As illustrated in Table 4.3, the effect of the view cone radius did not have a significant effect on the mean prey population. The only significant differences appeared with changes in the mortality parameters, where the prey population tended to grow marginally with an increase in radius. This indicates that the prey population may gain a slight advantage in a system where there is a predator bias (high d_p , b_h and low d_h) if they are given greater spatial awareness. As with the simple model, the prey population never went extinct.

Parameter Radius/Value	b_p			d_p		b_h		d_h	
	0.2	0.5	0.8	0.2	0.8	0.2	0.8	0.2	0.8
2	0.9867	0.9742	0.7712	0.9760	0.6409	0.9744	0.9063	0.4000	0.9762
3	0.9851	0.9742	0.7687	0.9754	0.6545	0.9746	0.8900	0.4051	0.9760
4	0.9865	0.9741	0.7696	0.9761	0.6744	0.9754	0.8967	0.4229	0.9760
5	0.9871	0.9748	0.7626	0.9675	0.6964	0.9748	0.9492	0.4490	0.9760

Table 4.3: *The effect of the view radius (r_v) on the prey population for different parameter combinations in the PPPE model.*

Effects of parameter variations on predators

The rules specific to predator individuals are not as dependent on the surrounding prey population, except that predators can only be born in cells where prey has died. Also, not every predator in the neighbourhood of a kill is eligible for reproduction, increasing pressure on the predator population. As in the simple model, the rule for predator death is completely independent from all other birth and death processes in the CA, therefore the value for d_h describes the average number of predators that are removed from the system per time step, regardless of position.

The proportion change in the mean predator population given a variation in a single parameter's value is illustrated in Figure 4.6. The population mean with respect to the base case, as

stipulated in §4.2.3, is indicated by the purple marker. As with the prey population, a decrease in d_p and an increase in d_h , had the greatest effect on the mean predator population, with an increase in b_h also having a marginal effect.

As illustrated in Table 4.4, there was little correlation between the view cone radius and the mean predator population. The only constant trend was for $d_p = 0.8$, where the predator population decreased with an increase in the radius, indicating that they had a marginal disadvantage if they were given a greater spatial awareness. It should be noted that the radius did not affect predator extinction, as for all four radii (with each radius having 91 tests) 58 tests lead to predator extinction. Table 4.5 confirms that, as seen in Figure 4.6, the predator population is more sensitive to d_p , b_h and d_h .

Parameter Radius/Value	b_p			d_p		b_h		d_h	
	0.2	0.5	0.8	0.2	0.8	0.2	0.8	0.2	0.8
2	0.0015	0.0009	0.0008	0.0007	0.0524	0.0007	0.0129	0.1215	0.9762
3	0.0019	0.0009	0.0007	0.0007	0.0505	0.0007	0.0160	0.1238	0.9760
4	0.0016	0.0008	0.0007	0.0007	0.0474	0.0007	0.0151	0.1244	0.9760
5	0.0014	0.0009	0.0007	0.0007	0.0443	0.0007	0.0057	0.1227	0.9760

Table 4.4: *The effect of the view radius (r_v) on the predator population for different parameter combinations in the PPPE model.*

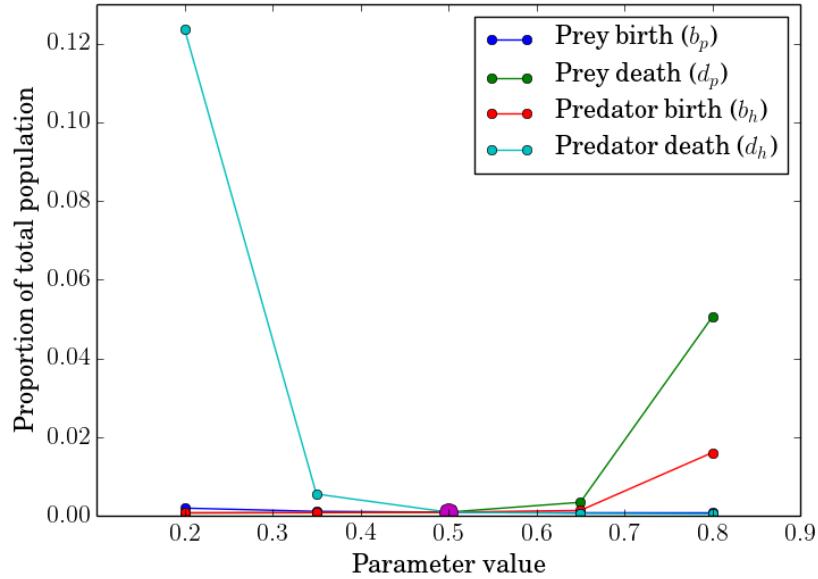


Figure 4.6: *The effect of single parameter variations on the mean predator population in the PPPE model.*

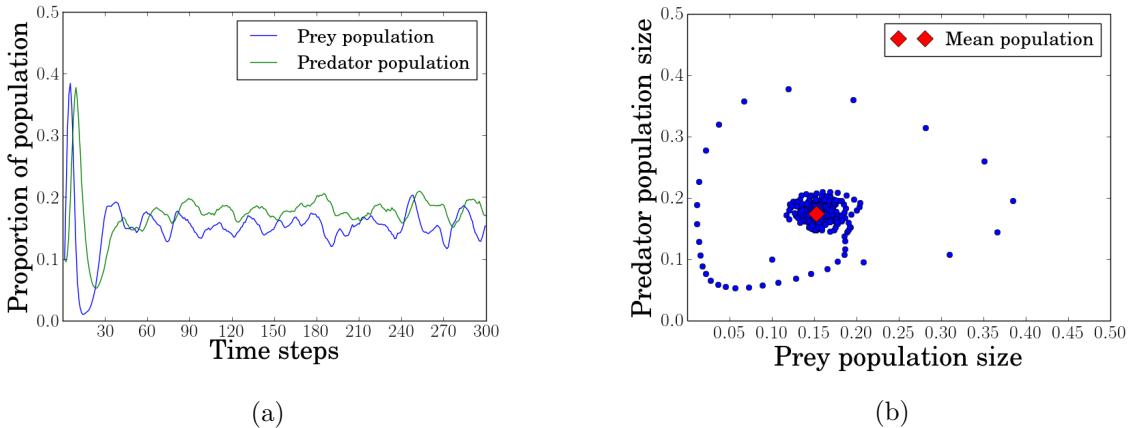
Parameter value	b_p	d_p	b_h	d_h
0.2	72	108	100	52
0.5	72	76	76	80
0.8	88	48	56	100

Table 4.5: *The number of predator extinctions per parameter value for the PPPE model.*

Review of the PPPE model

Comparing the results obtained from the PPPE model with those of the simple model yield many differences. In contrast to the simple model, which was particularly sensitive to variations in the reproduction parameters, the PPPE model is more sensitive to variations in the mortality parameters. A similarity between the models is that the predator population is at greater risk, again displaying heavy bias towards the prey population.

If a set of parameter values is chosen that would most favour the predator population, a local maximum mean predator population as well as a local minimum mean prey population is found. The system, despite its large initial spike, is stable and the populations oscillate in a manner that is similar to the Lotka-Volterra equations, as illustrated in Figure 4.7(a). The oscillations are also relatively small, as the deviation from the population mean is small after the thirtieth time step (see Figure 4.7(b)). The progression of the PPPE model algorithm if rendered graphically is illustrated in Figure 4.8. The clusters behave in a much more wave-like fashion than the simple model due to the spatial awareness and movement. Predator extinction usually takes place early in the runs, with extinction occurring within the first fifty time steps as illustrated in Figure 4.9.

Figure 4.7: (a) *The population dynamics and (b) the phase graphs for the PPPE model with parameters that favour predators.*

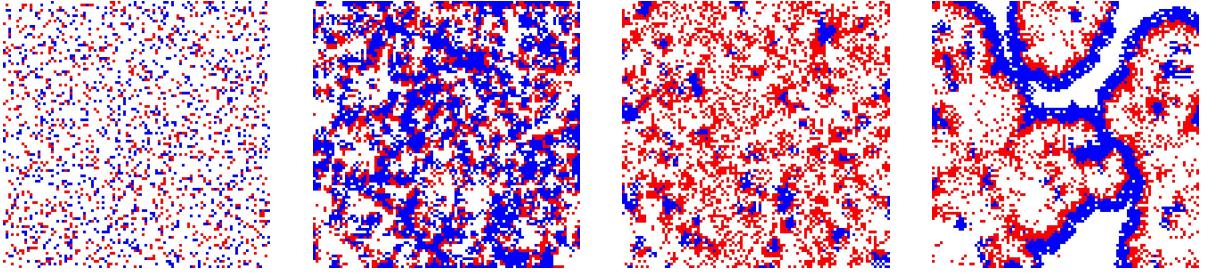


Figure 4.8: *Snapshots of the progression of the PPPE model for times $t = 1$, $t = 5$, $t = 10$ and $t = 35$ for a 100×100 array with parameters $[r_v = 3, 0.2, 0.8, 0.8, 0.2]$. Red represents the predators and blue the prey.*

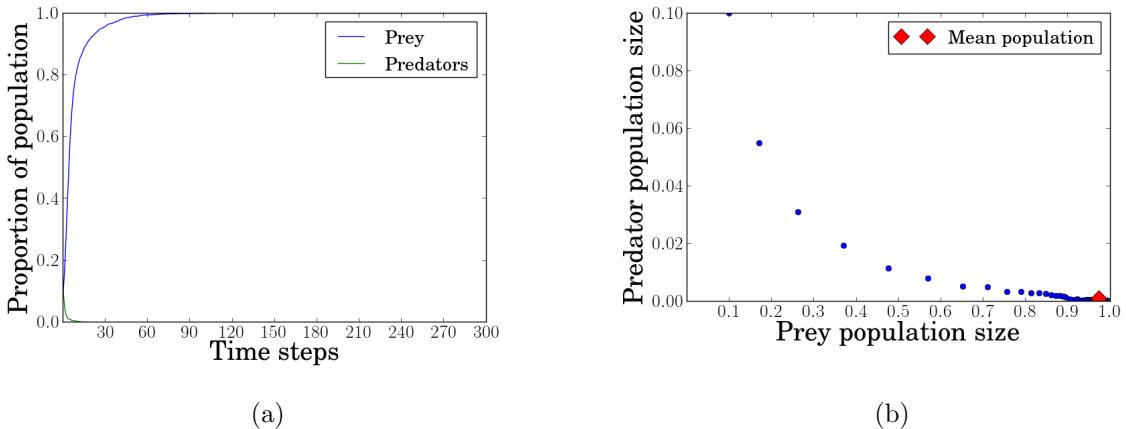


Figure 4.9: (a) *The population dynamics* and (b) *the phase graphs* for the PPPE model with the parameters set to the base case displayed rapid predator extinction.

4.2.4 DSAM model

For the DSAM model, which is based on the PPPE model, a second radius parameter is added that determines the size of the area used to calculate density sensitivity. There are two candidate functions proposed by Cattaneo *et al.* [4] that can map a probability value between 0 and 1 for density sensitivity: the cosine function in equation (3.6) and the exponential function in equation (3.7). The predator death parameter, d_h , is also changed from being a continuous value in $[0, 1]$ to a discrete value $\alpha > 0$. The six parameters are

- r_d , the radius of the density sensitivity area,
- r_v , the radius of the view cone as illustrated in Figure 3.6,
- b_p , the probability that a prey individual does not reproduce,
- d_p , the probability that a prey individual survives a predatory attack,
- b_h , the probability that a predator individual reproduces, and
- α , the maximum age in time steps of a predator.

The parameters may be divided into spatial awareness related, r_d and r_v , reproduction related, b_p and b_h , and mortality related, d_p and α . A total of 1055 parameter combinations were used during testing with a value set of $[0.2, 0.35, 0.5, 0.65, 0.8]$ for the probabilities (b_p, b_h, d_p, α) ,

[1, 2, 3] for the density radius r_d and [2, 3, 4, 5] for the view radius r_v , with the parameter values [$r_d = 2, r_v = 3, b_p = 0.5, d_p = 0.5, b_h = 0.5, \alpha = 20$] set as the base case for all experiments. The DSAM model utilising the cosine function for density sensitivity will be referred to as the cDSAM model, whereas the model utilising the exponential function will be referred to as the eDSAM model. As a result of having two functions for density sensitivity, it is necessary to observe the effects of parameter changes on both the cDSAM and eDSAM models. As the scale of d_h and α differ, the following mapping for α to the parameter values is made on the x -axis of all figures:

Parameter value	0.20	0.35	0.5	0.65	0.8
α	10	15	20	25	30

Table 4.6: *The parameter values mapped to different values for α .*

Effects of parameter variations on prey

The proportion change in the mean prey population given a variation in a single parameter's value is illustrated in Figures 4.10. The population mean with respect to the base case, as stipulated in §4.2.4, is indicated by the purple marker. For both models, the effects of parameter variations above 0.5 were very similar. The prey population had a lower carrying capacity in the eDSAM model than the cDSAM model, as the probability for death, given the same b_p values, would be higher for the eDSAM model, as illustrated in Figures 3.9a and 3.9b. This may also explain why there was a marginal difference between the populations when considering a variation in the value for b_p . Unlike in the previous models, a higher b_p lead to a slight increase in the mean prey population, with $b_p = 0.65$ proving to be the best compromise as it induced predator extinction without severely limiting the carrying capacity. The parameters d_p , b_h and d_h all resulted in a decrease in the mean prey population as they increased.

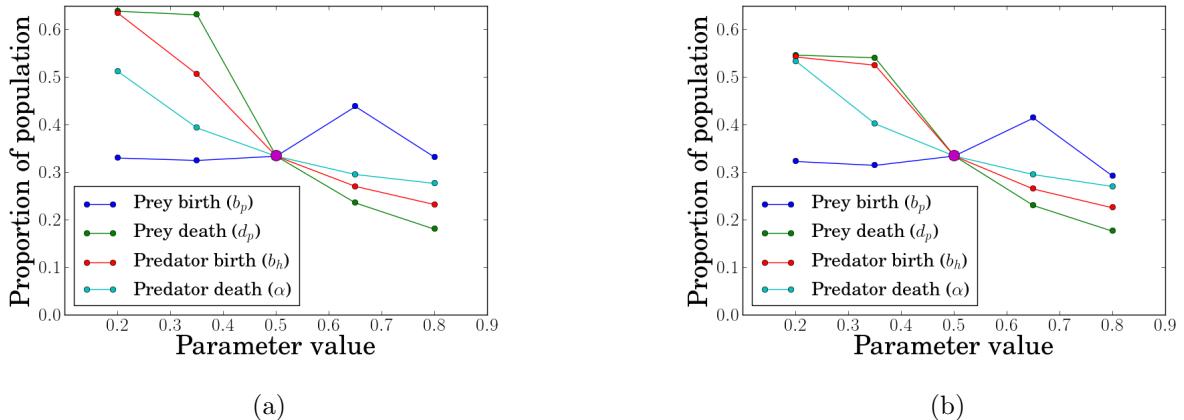


Figure 4.10: *The effect of single parameter variations on the mean prey population in the (a) cDSAM and (b) eDSAM models as a proportion of the total population.*

The DSAM models are the only models where prey extinction occurred. Three tests, all of which had parameter values favouring the predator population, lead to extinction of the prey population and, as a result, the predator population as well. Experimenting with some values for r_d yielded an interesting result, as setting $r_d = 1$ lead to predator extinction in the eDSAM model, while resulting in coexistence in the cDSAM model. This is due to the exponential function yielding much higher probabilities of death than the cosine function for $r_d = 1$, leading to a less dense prey population, making it more difficult for the predators to remain in the

neighbourhood of prey individuals. The difference in probabilities between the functions are illustrated in Figures 3.9a and 3.9b for $r_d = 2$. Changes in the prey population for different values of r_d are given in Table 4.7.

r_d	cDSAM	eDSAM
1	0.3538	0.4615
2	0.3344	0.3349
3	0.3342	0.3347

Table 4.7: *The mean prey population for different values of r_d in the cDSAM and eDSAM models.*

The added density sensitivity did appear to have an effect on the prey population when observing variations in the view radius (see Table 4.8). The PPPE model displayed only small changes in the prey population for an increase in r_v given a system with coexistence. The DSAM models showed a noticeable increase in the prey population as the view radius increased, indicating that greater spatial awareness had a positive effect on prey evasion.

r_v	cDSAM	eDSAM
2	0.3202	0.3154
3	0.3344	0.3349
4	0.3526	0.3575
5	0.3757	0.3806
10	0.5231	0.4789

Table 4.8: *The mean prey population for different values of r_v in the cDSAM and eDSAM models.*

Effects of parameter variations on predators

The proportion change in the mean predator population given a variation in a single parameter's value is illustrated in Figure 4.11. The population mean with respect to the base case, as stipulated in §4.2.4, is indicated by the purple marker. Comparing Figure 4.11 with Figure 4.10, one may observe that the effects of the various parameter variations are nearly opposite for the respective populations, with variations benefiting one population adversely affecting the other. The predator population appeared to be the most sensitive to changes in prey parameters, as setting d_p marginally lower or b_p marginally higher resulted in eventual extinction.

Similarly to the simple and PPPE models, the majority of tests on the DSAM models lead to predator extinction. A total of 1154 out of 2110 tests for the cDSAM and cDSAM models combined lead to predator extinction, with the eDSAM model having marginally more extinctions at 600 versus 554. The mean predator population for the eDSAM model was marginally smaller when compared to the cDSAM model, as the higher probability of death with the exponential function lead to a less dense prey population, limiting the predators' ability to feed and reproduce. An increase in the value of r_d caused notable growth in the predator population (see Table 4.9). This indicates that predators prefer prey that can live in more densely populated clusters, which is expected.

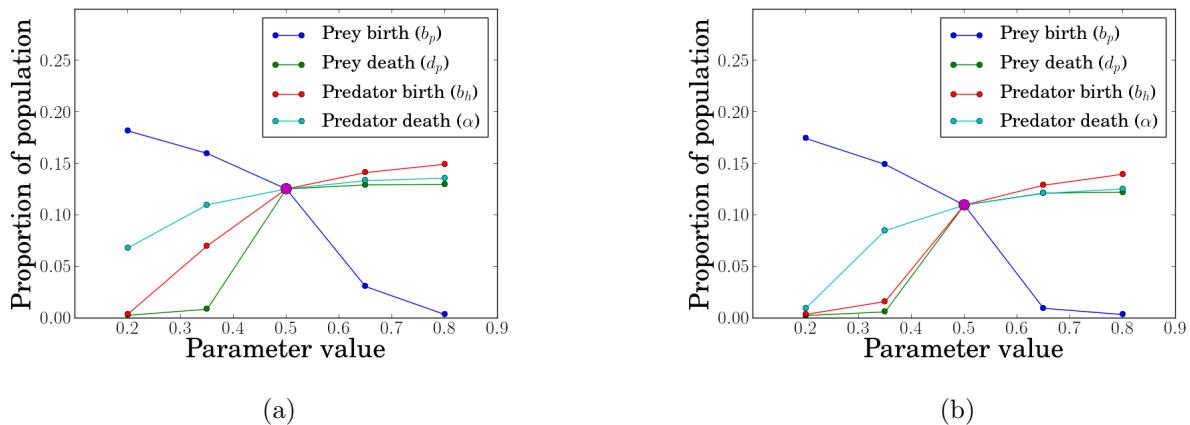


Figure 4.11: The effect of single parameter variation on the mean predator population in the (a) cDSAM and the (b) eDSAM model as a proportion of the total population.

r_d	cDSAM	eDSAM
1	0.1036	0.0163
2	0.1255	0.1098
3	0.1308	0.1218

Table 4.9: The mean predator population for different values of r_d for the cDSAM and eDSAM models.

Unlike the prey population, the view radius appeared to have little to no effect on the predator population. Variations in the viewing radius had an insignificant effect in both models for lower values, with the predator population in both models eventually decreasing as the view radius increased to ten and higher (see Table 4.10). This indicates that, given the same view radius, prey evasion is more effective than predator pursuit in the DSAM models. This does correlate with reality, as greater spatial awareness will give prey animals more time to flee.

r_v	cDSAM	eDSAM
2	0.1251	0.1126
3	0.1255	0.1098
4	0.1286	0.1099
5	0.1284	0.1066
10	0.0695	0.0507

Table 4.10: The mean predator population for different values of r_v in the cDSAM and eDSAM models.

Review of the DSAM model

The DSAM models are the only models that had stable coexistence with the base case parameter values. As with the simple and PPPE models, it reacted more with variations to one side of the base case, favouring values less than 0.5 for the probabilities. It did not favour any particular class of parameters, with density sensitivity only having a marginal effect on both populations. The age-adjusted mortality did appear to strengthen the predator population, as increases in α would help the predators survive for longer during random walks. The added complexities did

appear to have an effect on spatial awareness for moderate values, benefiting the prey population.

Since the base case resulted in stable coexistence, setting the parameters for the cDSAM model to the base case yields the stable oscillations as illustrated in Figure 4.12(a). The system quickly builds up to its stable prey population and oscillates closely around this point, as illustrated in Figure 4.12(b). The progression of the cDSAM model algorithm if rendered graphically is illustrated in Figure 4.13. The output is similar to that of the PPPE model given the same parameters, except for more empty cells being present in the prey wave as they die due to density sensitivity. It should be noted that the wave formation may be eliminated by changing only three parameter values, yielding a progression similar to that seen in the simple model, as illustrated in Figure 4.14.

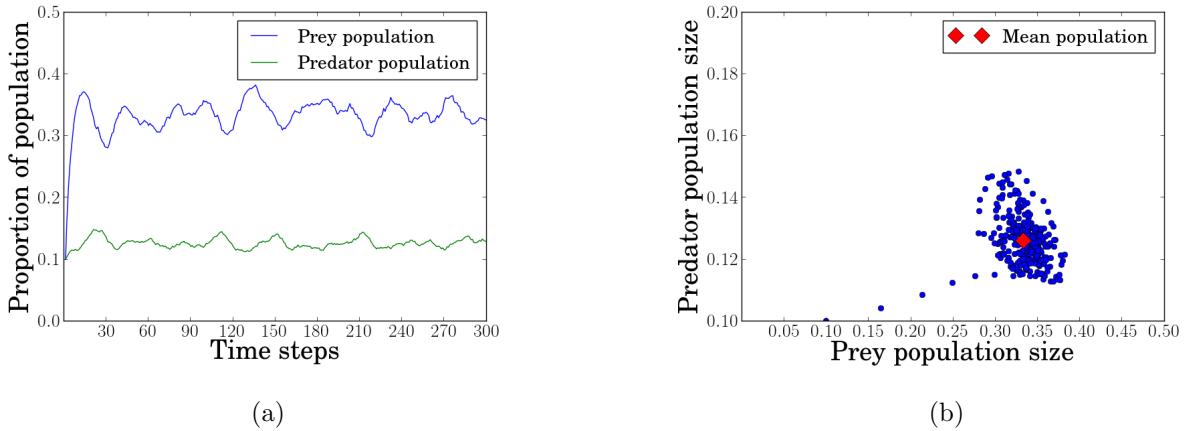


Figure 4.12: (a) The population dynamics and (b) the phase graph for the DSAM model with the parameters set to the base case.

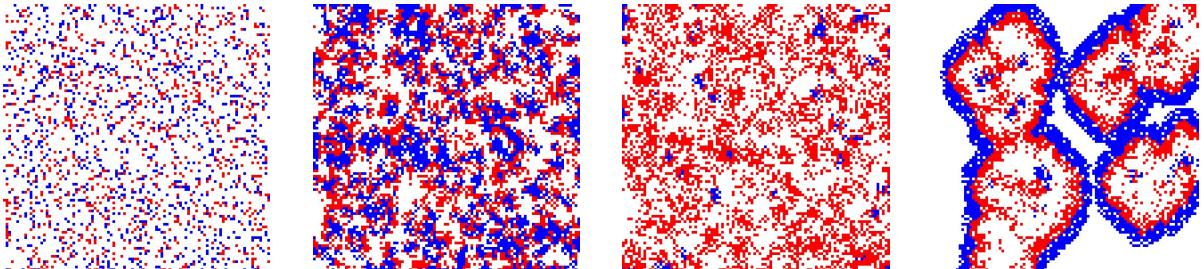


Figure 4.13: Snapshots of the progression of the cDSAM model for times $t = 1$, $t = 5$, $t = 10$ and $t = 35$ for a 100×100 array with parameters $[r_d = 2, r_v = 3, 0.2, 0.8, 0.8, \alpha = 20]$. Red represents the predators and blue the prey.

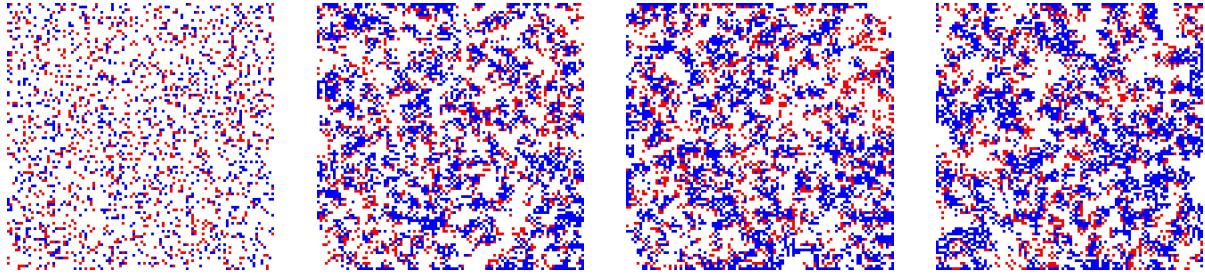


Figure 4.14: *Snapshots of the progression of the cDSAM model for times $t = 1$, $t = 5$, $t = 10$ and $t = 35$ for a 100×100 array with parameters $[r_d = 2, r_v = 3, 0.5, 0.5, 0.5, \alpha = 20]$. The wave formation has been subdued. Red represents the predators and blue the prey.*

4.3 Comparisons with existing models and array size

Comparison of the CA models with other mathematical or simulation models was done by observing pattern formations in the different types of models. Spatial awareness in the form of predator pursuit and prey evasion has been observed to form a spiralling of groups of individuals in PDE models by Li *et al.* [23] and in agent-based simulations by Hawick *et al.* [18].

The appearance of a defensive spiral in the prey population is a phenomenon observed by Hawick *et al.* [18] in an agent-based simulation of a predator-prey interaction. The prey population tended to flank the predators that were in pursuit, forming spiral formations that may help a portion of the prey population to survive an attack. Progression of the cDSAM model with parameter set $[b_p = 0.2, d_p = 0.8, b_h = 0.8, \alpha = 20, r_v = 3, r_d = 2]$ on a 50×50 array is illustrated in Figure 4.15. The emergence of a defensive spiral is evident as the prey population (blue) curls around the predator population pursuing it (red). This is important as it shows that the simpler CA implementation of the predator-prey interactions behave in a similar fashion to agent-based simulations as well as PDE models in literature.



Figure 4.15: *The cDSAM model displaying the emergence of a defensive spiral over four consecutive time steps.*

The spirals may have a detrimental effect on the prey population if the edges of the spirals converge. Once the spiral has been broken to form a solid and expanding circular wave, the prey population risk extinction by becoming trapped on the edges of the array. A scenario where a circular wave forms is illustrated in Figure 4.16, where a defensive spiral very similar to the one in Figure 4.15 converged to form an unbroken circular wave. The parameter values for the scenario were $[b_p = 0.2, d_p = 0.8, b_h = 0.8, \alpha = 30, r_v = 3, r_d = 2]$. Although the result is consistent for all simulations with the before mentioned parameter values, the behaviour is not consistent in all simulations with different parameter values, as the progression of the cDSAM model in Figure 4.14 displays no wave or spiral formation.

The effect of the array size may have an influence on the population persistence since the development of a single expanding circular wave is less likely given a wider distribution of individuals. The progression of the cDSAM model with the same parameter values as in Figure 4.16 on a 100×100 array is illustrated in Figure 4.17, resulting in a multitude of defensive spirals. Forcing the model to create a circular wave by setting the initial configuration to a closed square inverted the result, with the 50×50 array breaking the ring and leading to coexistence.

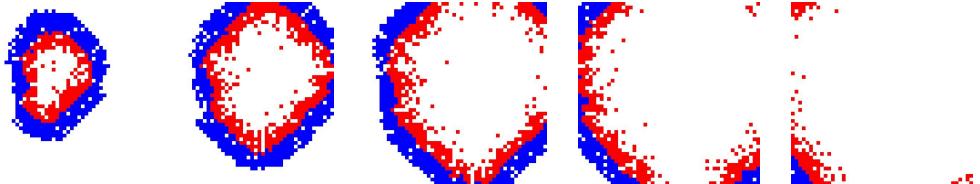


Figure 4.16: *The cDSAM model with a defensive spiral that converges to form a circular wave.*

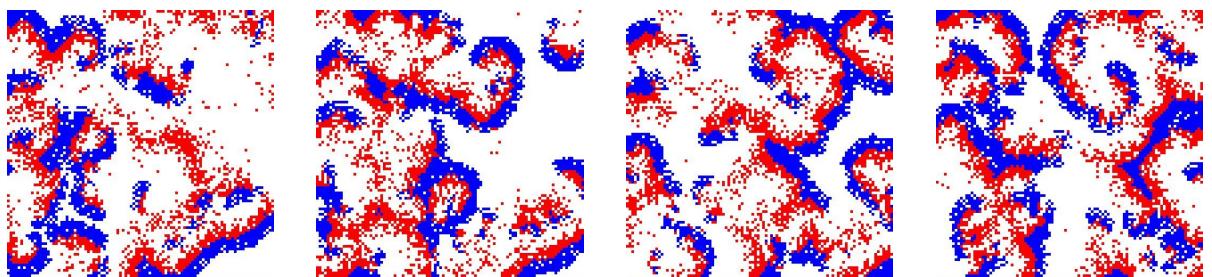


Figure 4.17: *The cDSAM model displaying the emergence of multiple defensive spirals over four consecutive time steps.*

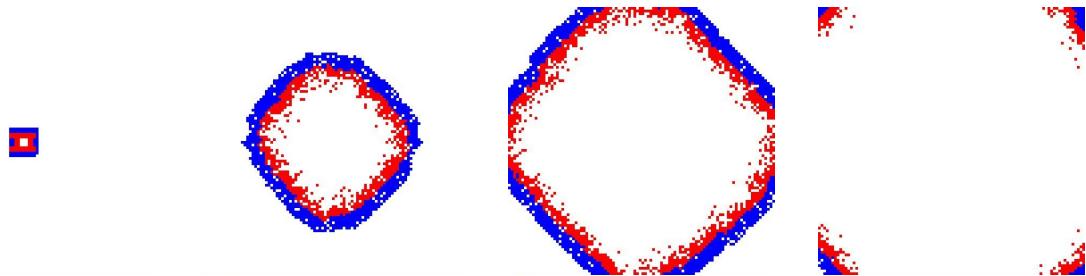


Figure 4.18: *The cDSAM model creates an unbroken ring given a square configuration over 90 time steps.*

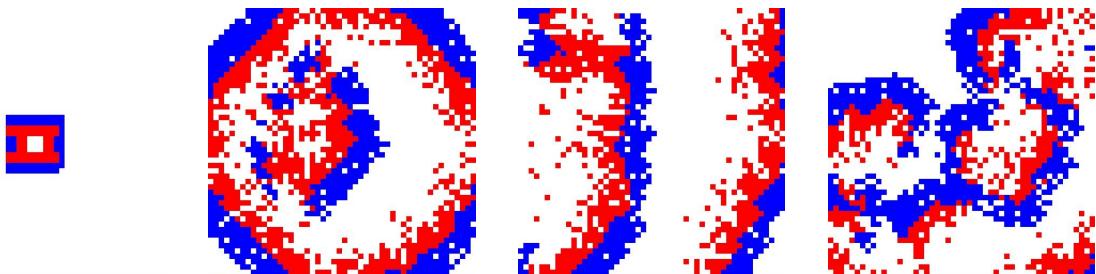


Figure 4.19: *The smaller cDSAM model breaks the ring and creates defensive spirals after 150 time steps.*

CHAPTER 5

Conclusion

This concluding chapter begins with a summary of the content in this project, followed by a brief description of the contributions made to literature during model development and analysis. The chapter closes with a discussion of potential ideas for future work and feasible expansions of the models developed.

5.1 Summary

A brief description of the core concepts involved, namely predator-prey interactions and cellular automata, were introduced in Chapter 1. A problem description was formulated questioning how these concepts may be combined to create a feasible model. It was determined that CAs modelling predator-prey interactions exist in literature, but that the analysis of these models were often lacking.

In Chapter 2, a literature review covering the history of mathematically modelling predator-prey interactions, development and applications of CAs and the rise of explicit agent-based simulations of predator-prey interactions was given. A base model for this project, developed by Cattaneo *et al.* [4], was also found to be very useful throughout the project. This was done in fulfilment of Objective I in §1.3.

Three predator-prey interaction CAs were developed and described in Chapter 3, each one building on the previous model. The simple model was developed in this study, utilising the structure of Hawick & Scogings [17] and the probability calculations from Cattaneo *et al.* [4]. The PPPE model, based on the work of Cattaneo *et al.* [4] with some modifications, implemented the predator pursuit and prey evasion mechanic, as required by Objectives II and III. The DSAM model, an extension of the PPPE model, added density sensitivity to the prey population and age-adjusted mortality to the predator population in fulfilment of Objective III.

The results obtained from parameter testing for the respective models were presented in Chapter 4. The effects that variations in parameter values may have on the prey and predator populations, respectively, were investigated in fulfilment of Objective IV. It was found that adding spatial awareness had a positive influence on the prey population, as they had a head start against any predators trying to pursue it. This is consistent with natural systems, as prey groups that have at least one individual as lookout tend to avoid predators more often e.g. meerkats or prairie dogs. It was also found that the respective models were not equally sensitive to the same

parameters, as a variation in the same parameter in two different models did not have the same effect. Some comparisons were also made to existing PDE and agent based simulation models, yielding similar results regarding the formation of spirals in models with spatial awareness. The reversal of results due to changes in initial configuration proved that population stability has dependencies not related to the rules of the model.

5.2 Contributions to literature

The main contributions of this project in the context of modelling predator-prey interactions by cellular automata, as far as could be determined by the author, are given in the this section.

Contribution 1 *A new and easy to implement CA to model the most basic interactions of predators and prey in a closed system.*

The simple model can be considered as a stand-alone model that was developed simply as a proof of concept. The rules are an amalgamation of many different implementations in literature and can subsequently be seen as a new model. Usage of the larger Moore neighbourhood is also a feature that sets the simple model apart from most other available models in literature.

Contribution 2 *Formalisation and analysis of the complete model developed by Cattaneo et al. [4].*

The model presented by Cattaneo *et al.* [4] is one of the most complete and complex predator-prey CAs in literature. The algorithm itself was presented poorly in literature, and the research did not go beyond developing the model. This project aimed to formalise Cattaneo's algorithm in the PPPE model and to additionally perform a thorough analysis on the effects that the parameters have on model stability and population sizes. It should be noted that there are two core differences between the PPPE model and the model developed by Cattaneo *et al.* [4]: The boundary conditions in the PPPE model are reflective whereas Cattaneo *et al.* [4] utilises a periodic boundary, and the predators embark on a random walk if no prey individuals are visible.

Contribution 3 *An age mechanic to be used in determining mortality probabilities.*

The DSAM models added a density sensitivity mechanic similar to the one proposed by Cattaneo *et al.* [4]. A new parameter, namely predator age, was added that may be used to more accurately model mortality rates. Predator individuals no longer die as part of a random process, but age each time step. Their age is then used to determine what their probability of death would be. Whilst age has been implemented in CAs in the past, the combination of all the mechanics present in the DSAM models, including age and the mechanics added to the PPPE model, are new to literature.

5.3 Future research

The models developed in this project are very generic and not specific to any existing system. The rules of the models, although complex in terms of what has been developed in literature, require many unrealistic assumptions and individuals behave unrealistically when compared

to a natural system. Subsequently, several recommendations can be made for possible future research.

Proposition 1: Investigate what effects the initial configuration, such as clustered or randomly distributed populations, may have on the system. The results in §4.3 indicated that population distribution had an effect on the system which requires further study.

Proposition 2: The effect that the array size may have as well as the effect of the various boundary conditions on the system was not investigated and warrants further study.

Proposition 3: Investigate whether the developed models can be applied to a real-world scenario with existing data. A key in benchmarking the models for use in real applications is to find a parameter set that closely models a known interaction so that it may be used as a feasible simulation for future applications.

Proposition 4: Create distinct individuals and not homogeneous species. Although the DSAM model did implement age for predator individuals, other characteristics and traits specific to certain individuals were not implemented. Adding gender and subsequently gender roles to individuals may add realism to models based on specific systems. Mechanics such as selective breeding and survival of the fittest may also be implemented in heterogeneous populations.

Proposition 5: Variations in terrain and the environment and their impact on species interactions may be added. An assumption in all developed models was that the space is homogeneous. Modifying the array as to mimic certain habitats, or including obstacles such as mountains, may contribute greatly to the perceived realism of the model.

Proposition 6: Develop hierarchical systems with an extended food chain. Most natural systems do not contain only two species that interact, but often dozens. Adding more species and their behaviour to the model would also contribute to the realism and applicability of the model to real-world scenarios.

List of References

- [1] BAULDRY W, 2011, *Introduction to Real Analysis: An Educational Approach*, John Wiley & Sons, United Kingdom.
- [2] BOCCARA N, ROBLIN O & ROGER M, 1994, *Automata network predator-prey model with pursuit and evasion*, Physical Review E, **50**(6), pp. 4531–4541.
- [3] BULLA M & RÁCZ ÉV, 2004, *Application of cellular automata in conservation biology and environmental management*, (Unpublished).
- [4] CATTANEO G, DENNUNZIO A & FARINA F, 2006, *A full cellular automaton to simulate predator-prey systems*, pp. 446–451 in *Cellular Automata*, pp. 446–451. Springer, Berlin.
- [5] CHAMPANDARD AJ, 2008, *A-life, emergent AI and S.T.A.L.K.E.R.: An interview with Dmitriy Iassenev*, [Online], [Cited April 28th, 2014], Available from <http://aigamedev.com/open/interviews/stalker-alife/>.
- [6] CHEN Q & MYNETT A, 2003, *Effects of cell size and configuration in cellular automata based prey-predator modelling*, Simulation Modelling Practice and Theory, **11**(7), pp. 609–625.
- [7] CHEN Q, MYNETT A & MINNS A, 2002, *Application of cellular automata to modelling competitive growths of two underwater species Chara Aspera and Potamogeton Pectinatus in Lake Veluwe*, Ecological Modelling, **147**(3), pp. 253–265.
- [8] CHOWDHURY DR, CHAUDHURI PP, BASU S & GUPTA IS, 1994, *Design of CAECC-cellular automata based error correcting code*, IEEE Transactions on Computers, **43**(6), pp. 759–764.
- [9] DONNART J & MEYER J, 1994, *A hierarchical classifier system implementing a motivationally autonomous animat*, From Animals to Animats, **3**(1), pp. 144–153.
- [10] EDELSTEIN-KESHET L & ERMENTROUT B, 1989, *Models for branching networks in two dimensions*, SIAM Journal on Applied Mathematics, **49**(4), pp. 1136–1157.
- [11] ERMENTROUT GB & EDELSTEIN-KESHET L, 1993, *Cellular automata approaches to biological modeling*, Journal of theoretical Biology, **160**(1), pp. 97–133.
- [12] FREZZA-BUET H & ALEXANDRE F, 2002, *From a biological to a computational model for the autonomous behavior of an animat*, Information Sciences, **144**(1), pp. 1–43.
- [13] FUENTES M & KUPERMAN M, 1999, *Cellular automata and epidemiological models with spatial dependence*, Physica A: Statistical Mechanics and its Applications, **267**(3), pp. 471–486.
- [14] GARDNER M, 1970, *Mathematical games: The fantastic combinations of John Conway's new solitaire game "life"*, Scientific American, **223**(4), pp. 120–123.

- [15] GAUSE GF, 1934, *Experimental analysis of Vito Volterra's mathematical theory of the struggle for existence*, Science, **79**(2036), pp. 16–17.
- [16] HASSELL M, MAY R, PACALA S & CHESSON P, 1991, *The persistence of host-parasitoid associations in patchy environments. A general criterion*, American Naturalist, **138**(3), pp. 568–583.
- [17] HAWICK K & SCOGINGS C, 2010, *A minimal spatial cellular automata for hierarchical predator-prey simulation of food chains*, Proceedings of the Proc. International Conference on Scientific Computing (CSC 10), WorldComp, Las Vegas, USA, pp. 75–80.
- [18] HAWICK K, SCOGINGS C & JAMES H, 2004, *Defensive spiral emergence in a predator-prey model*, Proceedings of the Complexity 2004, pp. 662–674.
- [19] HOPPENSTEADT F, 2006, *Predator-prey model*, Scholarpedia, **1**(10), p. 1563.
- [20] KARAFYLLODIS I & THANAILAKIS A, 1997, *A model for predicting forest fire spreading using cellular automata*, Ecological Modelling, **99**(1), pp. 87–97.
- [21] KONDO S & MIURA T, 2010, *Reaction-diffusion model as a framework for understanding biological pattern formation*, Science, **329**(5999), pp. 1616–1620.
- [22] KOT M, 2001, *Elements of mathematical ecology*, Cambridge University Press, United Kingdom.
- [23] LI ZZ, GAO M, HUI C, HAN XZ & SHI H, 2005, *Impact of predator pursuit and prey evasion on synchrony and spatial patterns in metapopulation*, Ecological Modelling, **185**(2), pp. 245–254.
- [24] LOTKA A, 1910, *Contribution to the theory of periodic reactions*, The Journal of Physical Chemistry, **14**(3), pp. 271–274.
- [25] LOTKA A, 1920, *Undamped oscillations derived from the law of mass action.*, Journal of the American Chemical Society, **42**(8), pp. 1595–1599.
- [26] LOTKA A, 1925, *Elements of physical biology*, Williams & Williams.
- [27] MILLS M & SHENK T, 1992, *Predator-prey relationships: The impact of lion predation on wildebeest and zebra populations*, Journal of Animal Ecology, **61**, pp. 693–702.
- [28] MURRAY J, 1976, *Spatial structures in predator-prey communities—a nonlinear time delay diffusional model*, Mathematical Biosciences, **31**(1), pp. 73–85.
- [29] QU S, CHEN Q & RECKNAGEL F, 2008, *Cellular automata based simulation of random versus selective harvesting strategies in predator-prey systems*, Ecological Informatics, **3**(3), pp. 252–258.
- [30] READY RATIOS, 2012, *Coefficient of variation*, [Online], [Cited October 12th, 2014], Available from <http://www.readyratios.com/reference/analysis/>.
- [31] SARKAR P, 2000, *A brief history of cellular automata*, ACM Computing Surveys, **32**(1), pp. 80–107.
- [32] SCOGINGS C & HAWICK KA, 2008, *Altruism amongst spatial predator-prey animats*, Proceedings of the ALIFE, pp. 537–544.
- [33] THOMAS R, 1798, *Malthus, an essay on the principle of population*, Masterworks of Economics, **1**(1), p. 26.

- [34] TOFFOLI T, 1984, *Cellular automata as an alternative to (rather than an approximation of) differential equations in modeling physics*, Physica D: Nonlinear Phenomena, **10(1)**, pp. 117–127.
- [35] VAN ZIJL L, 2014, Professor at *Department of Mathematical Sciences, Stellenbosch University*, [Personal Communication], Contactable at lvzijl@gmail.com.
- [36] VERHULST PF, 1838, *Notice sur la loi que la population suit dans son accroissement. correspondance mathématique et physique publiée par a*, Quetelet, **10(2)**, pp. 113–121.
- [37] VOLTERRA V, 1928, *Variations and fluctuations of the number of individuals in animal species living together*, Journal du Conseil, **3(1)**, pp. 3–51.
- [38] VON NEUMANN J, 1951, *The general and logical theory of automata*, Cerebral mechanisms in behavior, **1(1)**, pp. 288–326.
- [39] VON NEUMANN J & BURKS A, 1966, *Theory of self-reproducing automata*, University of Illinois press, New York.
- [40] WEGGE P, ODDEN M, POKHAREL C & STORAAS T, 2009, *Predator-prey relationships and responses of ungulates and their predators to the establishment of protected areas: a case study of tigers, leopards and their prey in Bardia National Park, Nepal*, Biological Conservation, **142(1)**, pp. 189–202.
- [41] WEISSTEIN E, 2014, *Cellular automaton*. From MathWorld—A Wolfram Web Resource, [Online], [Cited March 29th, 2014], Available from <http://mathworld.wolfram.com/CellularAutomaton.html>.
- [42] WHITE R, ENGELEN G & ULJEE I, 1997, *The use of constrained cellular automata for high-resolution modelling of urban land-use dynamics*, Environment and planning B, **24**, pp. 323–344.
- [43] WOLFRAM S, 1986, *Cryptography with cellular automata*, Proceedings of the Advances in Cryptology — CRYPTO '85 Proceedings, pp. 429–432.
- [44] WOLFRAM S, 1986, *Theory and applications of cellular automata*, World Scientific, Singapore.
- [45] WOLFRAM S, 2002, *A New Kind of Science*, Wolfram Media, Champaign, Illinois.
- [46] YOUNG DA, 1984, *A local activator-inhibitor model of vertebrate skin patterns*, Mathematical Biosciences, **72(1)**, pp. 51–58.
- [47] ZHENG S, 1986, *A reaction-diffusion system of a predator-prey-mutualist model*, Mathematical Biosciences, **78(2)**, pp. 217–245.

APPENDIX A

Results tables

Prey					
Parameter/Value	0.2	0.35	0.5	0.65	0.8
b_p	0.8397464	0.9629192	0.8508092	0.677797333	0.525786667
d_p	0.852524	0.8509312	0.8508092	0.850574933	0.848121067
b_h	0.853309867	0.854942133	0.8508092	0.789961067	0.5183672
d_h	0.841210933	0.845226933	0.8508092	0.854417333	0.8563864
Predators					
Parameter/Value	0.2	0.35	0.5	0.65	0.8
b_p	0.023463733	0.0010202672	0.0008756	0.000846133	0.000840267
d_p	0.000769067	0.000827467	0.0008756	0.001	0.001105067
b_h	0.000719467	0.000783733	0.0008756	0.005328533	0.039954133
d_h	0.002555733	0.001392533	0.0008756	0.000656533	0.0005168

Table A.1: *The mean population size for the prey and predator populations in the simple model for different parameter values.*

Prey					
Parameter/Value	0.2	0.35	0.5	0.65	0.8
b_p	0.985137067	0.983669333	0.974203733	0.9308144	0.768666667
d_p	0.975368267	0.9754288	0.974203733	0.956954133	0.654534933
b_h	0.974580533	0.974397467	0.974203733	0.972216	0.8899504
d_h	0.405129333	0.948493467	0.974203733	0.976105867	0.9759504
Predators					
Parameter/Value	0.2	0.35	0.5	0.65	0.8
b_p	0.001925333	0.001098933	0.0009112	0.000744	0.000730667
d_p	0.000701867	0.000746533	0.0009112	0.003397867	0.0504864
b_h	0.000737867	0.000796	0.0009112	0.0013328	0.016028
d_h	0.1237656	0.005509733	0.0009112	0.0005856	0.0004432

Table A.2: *The mean population size for the prey and predator populations in the PPPE model for different parameter values.*

Prey					
Parameter/Value	0.2(10)	0.35(15)	0.5(20)	0.65(25)	0.8(30)
b_p	0.330615652	0.325514114	0.334429967	0.438937793	0.33225097
d_p	0.63939612	0.631759732	0.334429967	0.236101405	0.180768428
b_h	0.635475585	0.506559197	0.334429967	0.270849231	0.232667559
α	0.512684682	0.393866221	0.334429967	0.295958261	0.277182341
Predators					
Parameter/Value	0.2(10)	0.35(15)	0.5(20)	0.65(25)	0.8(30)
b_p	0.181847224	0.160045485	0.125469699	0.030764147	0.003622742
d_p	0.002795184	0.008694849	0.125469699	0.129437057	0.12974796
b_h	0.00400107	0.070171505	0.125469699	0.141353579	0.149418863
α	0.068237324	0.110070368	0.125469699	0.133502609	0.135975652

Table A.3: *The mean population size for the prey and predator populations in the cDSAM model for different parameter values. α values are in parentheses.*

Prey					
Parameter/Value	0.2(10)	0.35(15)	0.5(20)	0.65(25)	0.8(30)
b_p	0.323346133	0.3151272	0.3348544	0.4147872	0.292546667
d_p	0.547089067	0.5414856	0.3348544	0.2306552	0.1765328
b_h	0.5431648	0.525811467	0.3348544	0.2652976	0.2258344
α	0.534377867	0.402643733	0.3348544	0.295784	0.270634933
Predators					
Parameter/Value	0.2(10)	0.35(15)	0.5(20)	0.65(25)	0.8(30)
b_p	0.1744832	0.149345867	0.109795867	0.009639733	0.003534133
d_p	0.002610933	0.006130667	0.109795867	0.121437333	0.122299467
b_h	0.0036968	0.0161272	0.109795867	0.1290656	0.139766933
α	0.009712	0.085075467	0.109795867	0.121185067	0.1254832

Table A.4: *The mean population size for the prey and predator populations in the eDSAM model for different parameter values. α values are in parentheses.*

APPENDIX B

Contents of the disk accompanying the project

The disk attached contains several source and data files related to the project. The contents include the required installation files, source code for the algorithms and output data. `Visualise_CA.py` can be used to execute the different models.

Installation Files The software required to run the algorithms:

- `python-3.3.0.msi` – Install file for a 32-bit Windows version of PYTHON 3.3.0,
- `pygame-3.3.0.msi` – Install file for PyGame, a requirement to visualise the algorithm,
- `numpy-3.3.0.exe` – Numerical Python installer required to run the algorithms,
- `scipy-3.3.0.exe` – Scientific Python installer, a prerequisite for Matplotlib, and
- `matplotlib-3.3.0.whl` – Python plotting library required to plot graphs.

Algorithms The algorithms developed in Chapter 3:

- `Automata_Simple.py` – The CA implementation of the simple model as in §3.3,
- `Automata_PPPE.py` – The CA implementation of the PPPE model as in §3.4,
- `Automata_DSAM.py` – The CA implementation of the DSAM model as in §3.5, and
- `Visualise_CA.py` – An executable that allows the user to configure and run the different CAs with parameters of their choosing, displaying the progression graphically as well as giving output graphs.

Data files The output data used to compile the results in Chapter 4:

- `Simple_Output.xlsx` – Output data for the simple model,
- `PPPE_Output.xlsx` – Output data for the PPPE model,
- `cDSAM_Output.xlsx` – Output data for the cDSAM model, and
- `eDSAM_Output.xlsx` – Output data for the eDSAM model.

hello