

SIMULIERTE KREATIVITÄT AM BEISPIEL DER ENTSTEHUNG KOOPERATIVEN VERHALTENS

Diplomarbeit

zur Erlangung des Grades Diplom-Wirtschaftsingenieur
(FH)

Hochschule Fulda
University of Applied Sciences



vorgelegt von:	Henrik Schaake
Studienbereich:	Elektrotechnik und Informationstechnik
Matrikelnummer:	621010
Erstgutachter:	Prof. Dr. Timm Grams
Zweitgutachter:	Prof. Dr. Viviane Wolff

Zusammenfassung

Kooperatives Verhalten erweist sich als stabil, wenn es sich etabliert hat und kann sich dann im Selektionsprozeß gegen unkooperatives Verhalten behaupten. Die Etablierung von kooperativem Verhalten erfordert jedoch das Vorhandensein einer ganzen Gruppe kooperativ Gesinnter. Die Entstehung von kooperativem Verhalten gelingt somit nicht unter den Bedingungen eines globalen Wettbewerbs. Mit einer Computersimulation auf Grundlage des Gefangenendilemmas soll unter der Verwendung von evolutionären Algorithmen die Entstehung kooperativen Verhaltens untersucht werden. Dazu wird der Mechanismus der Territorialität eingeführt. Mit diesem Mechanismus gelingt es in der Simulation kooperatives Verhalten entstehen zu lassen.

Inhaltsverzeichnis

Abbildungsverzeichnis	IV
Tabellenverzeichnis	V
1 Einleitung	1
2 Grundlagen	3
2.1 Simulierte Kreativität	3
2.2 Entstehung der Kooperation	7
3 Das Modell	12
3.1 Modellierung der Individuen	14
3.2 Evolutionsmechanismen	16
3.3 Bewertungsfunktion	17
3.4 Auswahloperatoren	17
3.5 Randbedingungen	18
4 Die Umsetzung	20
4.1 Auswahl des Interaktionspartners	20
4.2 Die Punkt-Mutation	21
4.3 Die lokale/globale Grenze	23
4.4 Weitere Änderungen	24
4.5 Die Steuerung	25
5 Versuche	28
5.1 Die Entstehung der Kooperation	28
5.2 Die gutmütigen Strategien	31
5.3 Das Quasi-Spezies-Modell	34
6 Schlussbetrachtung	39
Literaturverzeichnis	41
Eidesstattliche Erklärung	43

A Anhang	i
A.1 Inhalt CD	i

Abbildungsverzeichnis

3.1	Torus	12
3.2	Nachbarschaft/Umgebung	13
3.3	Baumdarstellung einer Strategie	14
3.4	Automatenmodell der Pawlow-Strategie	16
4.1	Anzahl an Nachbarn in einer k-Nachbarschaft	20
5.1	Spielfeldmatrix, Vergleich lokal und global	30
5.2	Histogramm eines Simulationslaufs mit der Mutationswahrscheinlichkeit $p=0,001$	36
5.3	Spielfeldmatrix	37
5.4	Histogramm eines Simulationslaufs mit der Mutationswahrscheinlichkeit $p=0,05$	38

Tabellenverzeichnis

2.1	Auszahlungsmatrix des Gefangenendilemmas nach Axelrod	7
3.1	Darstellung der Basischaraktere	15
3.2	Interaktion mit Pawlow	17
4.1	Eingabeparameter der Simulation	25
4.2	Farbcodierung der Basischaraktere	26
4.3	Informationen zu Wesen	26
5.1	Parameterwahl zum Vergleich von lokaler und globaler Grenze	30
5.2	Fehlerhafte Interaktion	32
5.3	Parameterwahl zur Untersuchung der Reaktion auf Interaktionsfehler	32
5.4	Wechselseitige Ausnutzung	33
5.5	Gutmütige Varianten von Tit for Tat	33
5.6	Parameterwahl zur Untersuchung der Empfindlichkeit auf die Mu- tationswahrscheinlichkeit p	35

1 Einleitung

Die vorliegende Arbeit befasst sich mit simulierter Kreativität am Beispiel der Entstehung kooperativen Verhaltens. Kreativität beschreibt eine Schöpferkraft, die hier unter der Verwendung von evolutionären Algorithmen nachgebildet wird.

Mit Hilfe der evolutionären Algorithmen soll die Entstehung von kooperativen Verhalten untersucht werden. Die Forschung zur Entstehung kooperativen Verhaltens orientiert sich an der Frage:

Unter welchen Bedingungen entsteht Kooperation in einer Welt von Egoisten ohne zentralen Herrschaftsstab?[Axelrod, 2006, S. 3]

Kooperation ist in vielen verschiedenen Bereichen und Ebenen anzutreffen. In der Wirtschaft kooperieren vermeintliche Konkurrenten in Form von Kartellen, im Tierreich bilden sich Rudel aus Individuen. Auch auf Zellebene kooperieren Organismen zur Bildung von funktionsfähigen Zellen.

Dass kooperatives Verhalten sich gegen unkooperatives Verhalten behaupten kann, wurde bereits von Axelrod gezeigt[Axelrod, 2006]. Voraussetzung dafür ist allerdings, dass sich kooperatives Verhalten bereits etabliert hat. Die Entstehung von kooperativen Verhalten benötigt zusätzliche Mechanismen. In einer Welt des globalen Wettbewerbs kann das kooperative Verhalten keinen Fuß fassen.

Um der Frage nachzugehen wie Kooperation entsteht wird hier eine Computersimulation in der Programmiersprache Java aufgebaut, die das iterierte Gefangenendilemma zur Grundlage von Interaktionen hat. Das Gefangenendilemma kann auf viele Ebenen von Interaktionen angewendet werden.

Das zugrunde liegende Modell wird dabei möglichst einfach gehalten. Es geht dabei nicht darum bestimmte Interaktionen möglichst realitätsnah abzubilden, sondern ein möglichst einfaches Modell mit wenigen Prämissen zu entwickeln, welches hilft das Verständnis für die Ausbildung von kooperativen Verhalten zu verbessern.

In der Simulation wird der Mechanismus der Territorialität eingeführt. Es soll überprüft werden, wie sich dieser Mechanismus auf die Entstehung der Kooperation auswirkt und ob es damit möglich ist, die Entstehung von kooperativen Verhalten zu simulieren.

In der Arbeit werden zunächst in Kapitel 2 die Grundlagen zu evolutionären Algorithmen vermittelt und Forschungsergebnisse zur Entstehung kooperativen Verhaltens zusammengetragen.

Im folgenden Kapitel 3 wird das verwendete Modell der Simulation genauer beschrieben und in Kapitel 4 wird auf die Umsetzung der Simulation in der Programmiersprache Java eingegangen.

In Kapitel 5 wird sich mit Versuchen zur Entstehung der Kooperation beschäftigt. Dabei werden die Auswirkungen der Territorialität genauer untersucht.

2 Grundlagen

2.1 Simulierte Kreativität

Die simulierte Kreativität tritt in Form von evolutionären Algorithmen in Erscheinung. Evolutionäre Algorithmen sind Heuristiken, die sich an die biologische Evolution anlehnen. Sie können für die Lösung von Optimierungsproblemen eingesetzt werden, auch für solche, die analytisch nicht zu lösen sind.

Lebewesen sind vollendete Problemlöser. In der Vielzahl der Aufgaben, die sie bewältigen, übertreffen sie die besten Computerprogramme bei weitem - zur besonderen Frustration der Programmierer, die Monate oder gar Jahre harter geistiger Arbeit für einen Algorithmus aufwenden, während Organismen ihre Fähigkeiten durch den scheinbar ziellosen Mechanismus der Evolution erwerben. [Holland, 1992]

Aus diesem Zitat von John H. Holland lässt sich bereits erahnen, welche Möglichkeiten die Nutzung von evolutionären Mechanismen bieten. Mit evolutionären Algorithmen können Lösungen zu Problemen gefunden werden, die der Programmierer selber nicht überschauen kann. Die erhaltene Lösung selber kann in ihrer Struktur so beschaffen sein, dass sie nicht genau verstanden wird, aber dennoch Ihre Anforderungen erfüllt.

Bei evolutionären Algorithmen wird nicht wie bei anderen Optimierungsverfahren wie z. B. dem Greedy-Algorithmus oder der Hill-Climbing Technik nur eine Lösung betrachtet, von der aus weitere Lösungen erarbeitet werden. Evolutionäre Algorithmen nutzen eine ganze Population von Lösungen. Diese Lösungen stehen in Konkurrenz zueinander.

Durch eine Bewertungsfunktion wird die Qualität einer Lösung bestimmt. Bessere Lösungen werden wahrscheinlicher zur Erzeugung neuer Lösungen genutzt, während schlechtere Lösungen aus der Population herausfallen. Bei der Erstellung von neuen Lösungen kann eine Vielzahl von Variationsoperatoren eingesetzt werden. Dabei können an die Biologie angelehnte Verfahren wie Mutationen, sexuelle Reproduktion und Crossing Over genutzt werden aber auch andere kreative Verfahren, die nicht in der realen Welt vorkommen. Die Variationsoperatoren müssen

jedoch an das jeweilige Problem angepasst sein, damit neue gültige Lösungen entstehen. Durch die Variationen werden neue Lösungen entdeckt und untersucht.

Für die Erstellung von evolutionären Algorithmen sind folgende Punkte genauer zu untersuchen [Michalewicz/Fogel, 2000]:

- Repräsentation der Individuen (Lösungen)
- Bewertungsfunktion
- Variationsoperatoren
- Auswahloperatoren
- Initialisierung

Dabei muss beachtet werden, dass die Punkte nicht isoliert voneinander betrachtet werden können, sondern miteinander abgestimmt werden müssen. So können Variationsoperatoren beispielsweise nur sinnvoll gewählt werden, wenn die Repräsentation berücksichtigt wird.

Für effektive evolutionäre Algorithmen muss soviel wie möglich über das Problem eingebracht werden, ohne das Ganze zu überladen. Mit zu wenig Informationen geht Effektivität verloren, zu viele Informationen können vorweg Lösungen ausschließen.

Repräsentation der Individuen

Mit der Repräsentation wird die Codierung der Lösungen festgelegt. Dafür gibt es meist viele Möglichkeiten. Die Codierung sollte im Zusammenhang mit den weiteren Operatoren eine sinnvolle Verbindung zwischen einem Elter und dessen Nachkommen ermöglichen.

Bewertungsfunktion

Die Bewertungsfunktion bestimmt die Qualität der einzelnen Lösungen. Dabei sollte die optimale Lösung den höchsten Wert zugewiesen bekommen.

Für einen effektiven Algorithmus ist es nicht sinnvoll eine „Alles oder Nichts“-Bewertungsfunktion zu nutzen. Eine solche Bewertungsfunktion würde der optimalen Lösung den höchsten Wert zuweisen und allen anderen Lösungen die gleiche niedrigere Qualität. Dies führt zu einer Art Zufallsuche.

Die Bewertungsfunktion muss Informationen bereitstellen, welche die Suche führen. Dies kann beispielsweise eine Punktzahl sein, die eine Lösung erzielt.

Variationsoperatoren

Die Variationsoperatoren generieren neue Lösungen auf Grundlage der bestehenden Lösungen. Sie sind von der gewählten Repräsentation und dem Lösungsraum abhängig.

Je nach Problem können Operatoren mit einem, zwei oder auch mehr Elternteilen ausgewählt werden.

Bei binären Repräsentationen wird häufig eine Mutation eingeführt, wobei jedes einzelne Bit mit einer bestimmten Wahrscheinlichkeit kippt. So ist es möglich von jedem Zustand zu jedem anderen Zustand zu wechseln.

Auswahloperatoren

Mit den Auswahloperatoren wird die Zusammensetzung der nächsten Population festgelegt. Grundsätzlich kann zwischen deterministischen und stochastischen Verfahren unterschieden werden. Bei deterministischen Verfahren werden bei gleicher Population stets die gleichen Individuen für Eliminierung und Fortpflanzung selektiert. Bei stochastischen Verfahren hingegen werden Wahrscheinlichkeiten für die Selektion eingesetzt.

Ein Beispiel für ein stochastisches Auswahlverfahren ist die „Roulette Wheel Selection“. Dabei ist die Wahrscheinlichkeit für eine Auswahl eines Individuums proportional zum erzielten Ergebnis des Individuums, d. h. ein Individuum das doppelt so gut bewertet wurde wie ein anderes hat auch eine doppelt so große Wahrscheinlichkeit ausgewählt zu werden im Vergleich zu dem anderen Individuum.

Ein deterministisches Auswahlverfahren hingegen könnte beispielsweise die Auswahl der zehn höchstbewerteten Individuen sein.

Deterministische Auswahlverfahren sind in der Regel schneller in der Ausführung und führen zu einer schnelleren Konvergenz der Population.

Im weiteren Verlauf der Arbeit wird ein stochastische Auswahlverfahren eingesetzt.

Initialisierung

Bei Optimierungsproblemen kann mit einer geeigneten Initialisierung die Effektivität erhöht werden. Es können bereits bekannte Informationen eingebracht werden, was den Zeitbedarf verkürzt. Ebenso ist es möglich bereits existierende Lösungen in die Startpopulation zu integrieren.

Eine Zufallsinitialisierung kann für Bewertungszwecke des Algorithmus sinnvoll erscheinen.

Dynamisches Umfeld

Für dieses Projekts wird ein genetischer Algorithmus verwendet. Der genetische Algorithmus wird jedoch nicht auf ein statisches Optimierungsproblem angewendet, sondern auf eine dynamische Umgebung. In einem herkömmlichen Optimierungsproblem erzielt eine Lösung stets das gleiche Ergebnis. Hier ist das Ergebnis einer einzelnen Lösung jedoch abhängig von der Zusammensetzung der Population.

2.2 Die Entstehung der Kooperation

Das Gefangenendilemma

Die Fragestellung mit der die Forschung zur Entstehung von kooperativen Verhalten begann war:

Wann sollte eine Person bei fortlaufender Interaktion mit einer anderen Person kooperieren, und wann sollte sie sich selbstsüchtig verhalten?
[Axelrod, 2006]

Untersuchungsgegenstand ist das iterierte Gefangenendilemma. Das Gefangenendilemma beschreibt eine Interaktion mit zwei Teilnehmern. Es geht auf folgende Situation zurück: Zwei Verbrecher werden gefasst und getrennt voneinander befragt. Jeder der beiden hat die Option zu kooperieren und zu schweigen oder zu defektieren und gegen den anderen Verbrecher auszusagen. Kooperieren beide erhalten sie nur eine geringe Strafe. Defektiert nur einer der beiden, so kommt der defektierende frei und der kooperierende erhält die Höchststrafe. Defektieren beide erhalten sie nur eine geringe Strafmilderung.[Mehlmann, 1997, S. 87]

Das Spiel kann mit einer Auszahlungsmatrix verdeutlicht werden.

Row- /Column Player	Cooperate	Defect
Cooperate	R=3, R=3	S=0, T=5
Defect	T=5, S=0	P=1, P=1

Tabelle 2.1: Auszahlungsmatrix des Gefangenendilemmas nach Axelrod [Axelrod, 2006, S. 8]

R steht hierbei für „Reward for mutual cooperation“, S für „Sucker’s payoff“, T für „Temptation to defect“ und P für „Punishment for mutual defection“. Die Auszahlungsmatrix kann variiert werden. Dabei muss beachtet werden, dass die Reihenfolge beibehalten wird.

$$T > R > P > S$$

Die höchste Auszahlung die ein Spieler erhalten kann ist somit die Versuchung zu defektieren, gefolgt von der Belohnung für beiderseitige Kooperation. Danach kommt die Bestrafung für beiderseitige Defektion und zuletzt die Auszahlung des gutgläubigen Opfers.

Die nächste Prämisse für das Gefangenendilemma besagt, dass das Dilemma nicht durch gegenseitiges wechselndes Ausnutzen gelöst werden kann.

$$2 \cdot R > T + S$$

Wenn die beiden Spieler sich gegenseitig abwechselnd ausnutzen erhalten sie im Durchschnitt $\frac{T+S}{2}$ Punkte. Bei durchgehender beiderseitiger Kooperation könnten sie jedoch R Punkte erreichen.

Das ursprüngliche Gefangenendilemma wird nur eine Runde gespielt. Dadurch entsteht für einen rational handelnden Spieler kein Anreiz zum kooperieren. Jeder rationale Spieler würde hierbei defektieren. Unabhängig von der Wahl des zweiten Spielers bietet Defektion eine höhere Auszahlung. Defektiert der Gegenspieler, würde defektieren die Bestrafung für beiderseitige Defektion erbringen, was immernoch höher ist als die Auszahlung des gutgläubigen Opfers ($P > S$), die durch Kooperation erreicht werden würde. Kooperiert der Gegenspieler so kann ebenfalls durch Defektion eine höhere Auszahlung erwirkt werden als durch Kooperation ($T > R$).

Darin liegt das Dilemma. Zwei rational handelnde Spieler würden somit die Auszahlung P erhalten, obwohl die höhere Auszahlung R für beide Spieler erreichbar wäre.

Wird das Spiel über eine bekannte Anzahl von Runden gespielt, so gibt es für den rational handelnden Spieler in der letzten Runde keinen Anreiz für Kooperation, da eine weitere Spielrunde ausgeschlossen ist. Ein Ansatz der Spieltheorie leitet davon ab, dass somit keine Kooperation zustande kommen würde, da auch in der vorhergehende Runde kein Anreiz für Kooperation bestehe. Rekursiv betrachtet ergibt sich somit die Strategie des steten Defektierens.[Axelrod, 2006]

Somit ist es vorteilhaft, wenn das Spiel über eine unbekannt Anzahl von Runden gespielt wird. Dies kann dadurch geschehen, dass das Spiel über eine bestimmte Anzahl an Runden gespielt wird, die den Spielern nicht mitgeteilt wird. Eine andere Alternative besteht darin, das Spiel in jeder Runde mit einer bestimmten Wahrscheinlichkeit abubrechen.

Im iterierten Gefangenendilemma nach Axelrod werden Auszahlungen zu früheren Zeitpunkten höher bewertet als zu späteren Zeitpunkten. Dies geschieht da Spieler Erträge höher bewerten, wenn sie in naher Zukunft anfallen. Diese Erträge sind wertvoller als Erträge in ferner Zukunft, da diese bereits jetzt einen Vorteil erbringen und auch sicherer sind. Eine Interaktion zwischen zwei Teilnehmern kann beendet werden. Geschieht dies, sind keine weiteren Erträge zu erwarten. Erreicht wird die höhere Bewertung von Erträgen zu früheren Zeitpunkten durch einen Diskontparameter w , der die Gewichtung des nächsten Zuges beschreibt. Somit wird ein Zug i ($i = \{1, 2, 3, \dots\}$) mit w^{i-1} bewertet mit $0 < w < 1$.

Das Gefangenendilemma kann aufgrund der wenigen getroffenen Prämissen auf eine Vielzahl von Situationen angewendet werden. Dazu zählen Interaktionen von

Politikern im US Senat, bilaterale Beziehungen zwischen Staaten aber auch Vorgänge auf molekularer Ebene. Es ist nicht notwendig, dass die Auszahlungen absolut vergleichbar sind. Die Auszahlungen können von den beiden Spielern auch als verschieden aufgefasst werden. Es reicht bereits aus, wenn die Auszahlungen relativ zueinander geordnet werden können und den beschriebenen Prämissen folgen ($T > R > P > S$ und $R > \frac{T+S}{2}$).

Nicht erfasst durch das Gefangenendilemma werden Null-Summen-Spiele, wie Schach. Bei diesen Spielen bedeutet der Gewinn des einen den Verlust des anderen.

Axelrods Computerturniere

Um erfolgreiche Strategien für das Verhalten im Gefangenendilemma zu finden richtete Robert Axelrod zwei Computerturniere aus. Am ersten nahmen Vertreter der Fachrichtungen Psychologie, Wirtschaft, Politikwissenschaften, Mathematik und Soziologie teil, in der zweiten Runde konnte das Teilnehmerfeld sogar erweitert werden.

Das Turnier war rundenbasiert, jede Strategie wurde mit allen eingesendeten Strategien, sowie mit einer Zufallsstrategie gepaart. Die Strategien hatten ein Gedächtnis, so konnten alle bisherigen Resultate für die nächste Entscheidung verwendet werden. Im ersten Turnier wurde eine fixe Anzahl von Zusammentreffen je Paarung von 200 gewählt, im zweiten Turnier wurde eine Abbruchwahrscheinlichkeit genutzt um Endspielstrategien zu vermeiden.

Der Sieger beider Turniere ist die Strategie Tit For Tat (TFT). Die Strategie TFT benötigt nur ein Gedächtnis der Tiefe eins, sie muss sich lediglich an die letzte Aktion des Mitspielers erinnern können. Sie ist eine „nette“ Strategie. Eine Strategie wird als nett bezeichnet, wenn sie nicht als erstes defektiert. TFT beginnt mit Kooperation im ersten Zug und wählt als nächsten Zug den aktuellen Zug seines Gegenübers.

Das Ergebnis der Turniere besagt nicht, dass TFT immer die beste Strategie ist. Eine beste Strategie gibt es nicht, wenn der Zukunft genug Bedeutung beigemessen wird. Dies ist der Fall wenn der Diskontparameter w groß genug ist ($w > \frac{T-R}{T-P}$)¹.

¹Ein Folge von Belohnungen für Kooperationen (R) muss größer sein als die Versuchung zu defektieren (T) mit anschließenden Bestrafungen (P)[Axelrod, 2006]:

$$R + wR + w^2R + w^3R + \dots > T + wP + w^2P + w^3P + \dots \quad (2.1)$$

$$\frac{R}{1-w} > T + \frac{wP}{1-w} \quad (2.2)$$

$$w > \frac{T-R}{T-P} \quad (2.3)$$

Das Abschneiden einer Strategie ist immer abhängig von der Zusammensetzung der Population mit der interagiert wird. Bereits leichte Modifikationen an anderen Strategien, hätten für diese den Sieg bedeutet. Das Ergebnis ist allerdings ein Indiz dafür, dass TFT in einer Vielzahl von Umgebungen erfolgreich ist, wie auch weitere Untersuchungen von Axelrod zeigen.

Tit For Tat vereinigt einige wichtige Eigenschaften, die sich bei Axelrods Computerturnieren als wesentliche Erfolgsfaktoren für Strategien herausgestellt haben. TFT ist

- nett
TFT ist eine nette Strategie, sie defektiert nicht als erstes. In Axelrods Computerturnieren wurden die besten Ergebnisse von netten Strategien erzielt. Im ersten Turnier waren die besten acht Strategien nett, im zweiten Turnier waren die 15 besten bis auf eine Ausnahme nett.
- vergeltend
TFT ist eine vergeltende Strategie, sie lässt sich nicht ausnutzen. Auf eine Defektion wird ebenfalls mit Defektion reagiert.
- vergebend
TFT ist eine vergebende Strategie. Kooperiert der Interaktionspartner nach einer Defektion wieder, so wird die Kooperation erwidert. Ein Gegenbeispiel dafür ist der „konsequente Vergelter“ (s_{immerD}). Diese Strategie kooperiert bis zur ersten Defektion des Mitspielers und von diesem Zeitpunkt kooperiert sie nie mehr, unabhängig vom Verhalten des Gegenüber.
- einfach
TFT ist eine relativ einfache Strategie. Zu komplexe und gerissene Strategien, die versuchen Modelle des Gegenübers zu entwerfen oder mit Wahrscheinlichkeiten arbeiten, erscheinen ihrem Gegenüber häufig nicht vertrauensvoll oder gar zufällig. Im Gefangenendilemma ist es nicht wie in Nullsummen-Spielen von Vorteil die eigene Absicht zu verschleiern. Gerade die Transparenz ist bei TFT ein Anreiz zur Kooperation.
- nicht neidisch
TFT ist eine nicht neidische Strategie, sie erhält nie mehr Punkte als sein Gegenüber, maximal erhält TFT die gleiche Punktzahl. Anders als in Nullsummenspielen geht es nicht darum besser abzuschneiden als der Mitspieler oder diesen zu vernichten. Somit ist für die Bewertung des eigenen Spiels nicht etwa die Punktezahl des Gegners ausschlaggebend, sondern die Punktezahl, die man maximal in Interaktion mit der gegnerischen Strategie hätte erzielen

können. Wäre für den Spieler das Ziel nur besser als sein Gegenüber abzuschneiden, so müsste dieser die Strategie des steten Defektierens nutzen.

Die in dieser Arbeit verwendeten Strategien haben keine Informationen darüber, wie ihre Gegner abschneiden. Somit ist Neid für die hier verwendeten Strategien nicht von Bedeutung. Für die allgemeine Betrachtung der Erfolgsfaktoren einer Strategie sollte diese Eigenschaft jedoch nicht unterschlagen werden.

Dynamische Simulationen

In Axelrods Computerturnieren liegen statische Umgebungen vor. Die Population ist festgelegt durch die eingesendeten Lösungen und unveränderlich. Um dynamische Betrachtungen anzustellen, simulierte Axelrod zukünftige Generationen der Turnier-Population durch eine ökologische Simulation. Die Zahl der Nachkommen einer Strategie wurde proportional zu ihren erzielten Punkten gewählt. Das entspricht dem Prinzip des Überlebens der Tauglichsten.

Bei dieser Simulation ist deutlich zu sehen wie anfangs alle erfolgreichen Strategien wachsen, während nicht erfolgreiche Strategien relativ bald aussterben. Das Aussterben der schwachen Strategien beeinträchtigt kooperative Strategien wie TFT nicht. Kooperative Strategien erzielen untereinander hohe Punktzahlen. Nicht nette Strategien, deren Erfolg auf Ausbeutung beruht, stehen allerdings vor Problemen wenn die schwachen Strategien aussterben. Sie entziehen sich damit selbst ihrer Lebensgrundlage.[Axelrod, 2006]

Die Arbeiten von Axelrod regten zu weiteren dynamischen Simulationen an. In weiteren Simulationen wurden auch evolutionäre Ansätze verfolgt, beispielsweise in einer Simulation von Lomborg [Lomborg, 1996].

3 Das Modell

Ausgangspunkt ist eine Simulation von Grams, die Grundideen von Axelrods Computerturnieren mit dem Ansatz der Kugelspiele kombiniert. Ziel ist die Untersuchung der Evolution der Kooperation. Dabei wird ein evolutionärer Ansatz mit Hilfe eines evolutionären Algorithmus verfolgt.

Auf einem Spielfeld werden Individuen mit festen Strategien platziert. Diese Individuen spielen mit ihrer Nachbarschaft das iterierte Gefangenendilemma. Je nach Erfolg der Individuen können diese Nachwuchs erzeugen oder sterben.

Die Simulation hat Ähnlichkeit mit einem zellulären Automaten. Der Zellraum auf dem die Simulation stattfindet ist zweidimensional mit rechteckiger Gittergeometrie. Die Abmaße der Spielfeldmatrix ist quadratisch mit $n \cdot n$ Feldern. Auf einem Feld kann entweder ein Individuum angesiedelt sein, oder es kann frei sein.

Für das Spielfeld werden periodische Randbedingungen gewählt. Ober- und Unterseite, sowie linke und rechte Seite werden miteinander verbunden, so dass ein Torus entsteht wie in Abbildung 3.1 zu sehen.

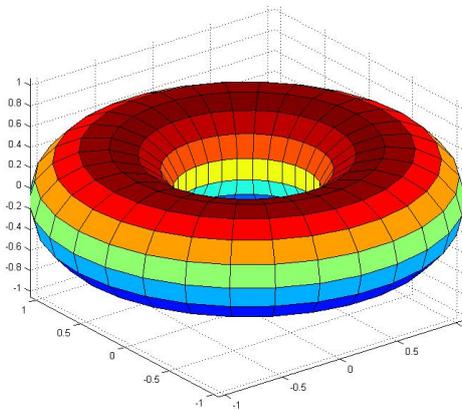


Abbildung 3.1: Torus

Jedes Feld hat eine k -Nachbarschaft bzw. k -Umgebung. Diese Nachbarschaft kann durch die Summennorm oder die Maximumnorm definiert werden.

Die Summennorm mit $k = 1$ wird auch als Von-Neumann-Nachbarschaft bezeichnet, da diese in Von-Neumanns Urversion eines zellulären Automaten verwendet wurde.[Gerhardt/Schuster, 1995]

Für ein Feld (i, j) ist die k -Nachbarschaft definiert durch

$$(i', j') \in U_k(i, j) \text{ mit } |i' - i| + |j' - j| \leq k \quad (3.1)$$

Die Maximumnorm beschreibt die Moore-Nachbarschaft (für $k = 1$) bzw. die erweiterte Moore-Nachbarschaft:

$$(i', j') \in U_k(i, j) \text{ mit } \max \{|i' - i|, |j' - j|\} \leq k \quad (3.2)$$

[Grams, 2007]

In Abbildung 3.2 sind beide Nachbarschaften abgebildet. Die Simulation verwendet die Maximumnorm.

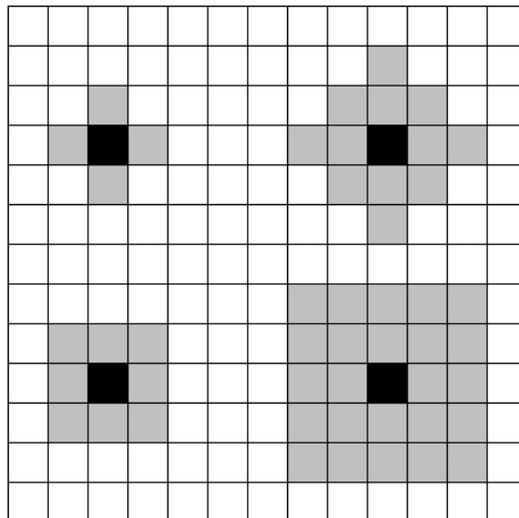


Abbildung 3.2: Nachbarschaften (grau) von Individuen (schwarz), links oben ist die Von-Neumann-Nachbarschaft zu sehen, rechts oben die Nachbarschaft nach Summennorm mit $k=2$, links unten ist die Moore-Nachbarschaft und rechts unten die erweiterte Moore-Nachbarschaft mit $k=2$ abgebildet

Die Entwicklung der Individuen verläuft asynchron. In einem Spielzug wird ein zufälliges Feld (i, j) auf der Spielmatrix ausgewählt, sowie ein Feld $(i', j') \in U_k(i, j)$ aus der k -Nachbarschaft nach dem Zufallsprinzip.

Ist das Feld(i, j) von einem Individuum besetzt, wird geprüft ob auch Feld(i', j') besetzt ist. Ist dies der Fall kommt eine Interaktion zustande. Bei der Interaktion spielen die Teilnehmer eine Runde des Gefangenendilemmas. Anschließend wird mit Hilfe der Auswahloperatoren geprüft, ob das Individuum auf dem Feld (i, j) in der Population verbleibt oder stirbt.

Ist das Feld(i, j) frei, gilt festzustellen ob das Feld(i', j') besetzt ist. Ist das Feld(i', j') besetzt, so kann dieses einen Nachkommen zeugen, sofern es von den Auswahloperatoren selektiert wird.

3.1 Modellierung der Individuen

Jedes Individuum besitzt eine feste Strategie, die sich im Laufe des Lebens des Individuums nicht verändert. Eine Strategie legt den nächsten Zug auf Grundlage von vergangenen Zügen fest. Für diese Simulation werden für die Entscheidungsfindung nur die Züge des Gegners berücksichtigt, nicht die eigenen Züge. Zusätzlich wird sich auf deterministische Strategien beschränkt.

Um eine Strategie umzusetzen benötigt jedes Individuum ein Gedächtnis je Interaktionspartner. Die Gedächtnistiefe wird auf drei Züge festgelegt. Somit besteht eine Strategie aus 2^3 Entscheidungen der Form kooperieren oder defektieren. Die Strategien können als 8 Bit Worte dargestellt werden (zu sehen in Abbildung 3.3), wobei eine 0 für kooperieren und eine 1 für defektieren steht.

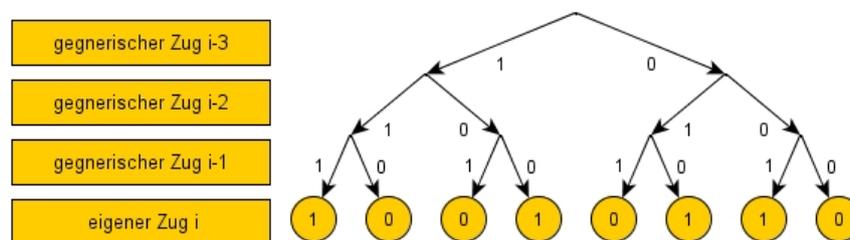


Abbildung 3.3: Baumdarstellung einer Strategie

Folglich gibt es 2^8 mögliche Strategien. Aus diesen Strategien sind bekannte Basischaraktere in Tabelle 3.1 zu sehen.

Zwei der in Tabelle 3.1 dargestellten Basischaraktere können mit dem begrenzten Mitteln nicht korrekt nachgebildet werden. Der konsequente Vergelter bräuchte ein

Historie:	111	110	101	100	011	010	001	000
<i>STitForTat</i>	1	0	1	0	1	0	1	0
<i>SVergelter</i>	1	1	1	1	1	1	1	0
<i>SImmerK</i>	0	0	0	0	0	0	0	0
<i>SPawlow</i>	1	0	0	1	0	1	1	0
<i>SImmerD</i>	1	1	1	1	1	1	1	1
<i>SHinterlist</i>	0	1	0	1	0	1	0	1

Tabelle 3.1: Darstellung der Basischaraktere [Grams, 2007]

unendlich langes Gedächtnis um ihn in dieser Form korrekt nachzubilden. In dieser Darstellung vergibt der konsequente Vergelter eine Defektion nach dreimaliger Kooperation.

Die Strategie Pawlow ist eine nette Strategie nach dem Grundsatz „win stay, loose shift“. Sie kooperiert nur bei Übereinstimmung mit dem Gegenspieler, ansonsten defektiert sie. Um diese Strategie korrekt abzubilden wird die Information über den eigenen Zug benötigt. Da diese Information in der Simulation nicht verfügbar ist, ist die Pawlow Strategie nur ansatzweise nachzubilden. In Abbildung 3.4 ist ein Automatenmodell von der Pawlow-Strategie zu sehen, wie sie hier vertreten ist.

Abbildung 3.4 zeigt, dass beginnend mit der Historie 000 die ersten vier Entscheidungen korrekt getroffen werden, die Strategie dann aber von der Pawlow-Strategie abweicht.

Besonders auffällig ist, dass diese Strategie somit nicht in der Lage ist gutmütige Strategien konsequent auszunutzen. In Tabelle 3.2 ist das Zusammentreffen der Strategie Pawlow mit einer potentiell ausnutzbaren Strategie zu sehen. Nach einer einzelnen Defektion im ersten Zug kooperiert die gutmütige Strategie fortan. Pawlow müsste als Reaktion konsequent ab dem 2. Zug defektieren. Ab dem vierten Zug befindet sich das Gedächtnis jedoch wieder im Zustand „000“ worauf die Kooperation wieder aufgenommen wird.

Ebenso verhält sich die hier gezeigte Pawlow-ähnliche Strategie im Zustand 111 nicht korrekt. Während die Pawlow-Strategie beim Zusammentreffen mit *sImmerD* abwechselnd kooperieren und defektieren würde, bleibt die hier abgebildete Strategie beim Defektieren sobald eine Historie von „111“ erreicht ist.

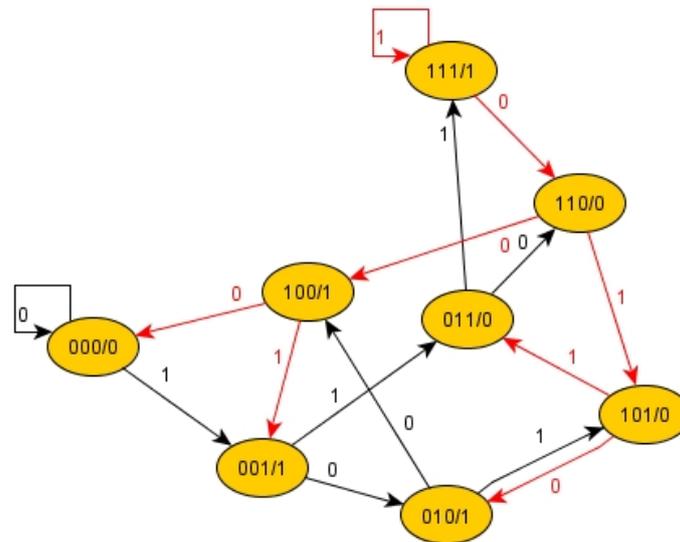


Abbildung 3.4: Automatenmodell der Pawlow-Strategie (Gedächtnis/nächster Zug). Der Pfeil beschreibt den gegnerischen Zug. Die roten Pfeile stehen für Züge auf die nicht entsprechend der Pawlow-Strategie reagiert wird.

3.2 Evolutionsmechanismen (Variationsoperatoren)

Als Evolutionsmechanismus wird die Punktmutation verwendet. Die Mutationswahrscheinlichkeit p gibt dabei an, mit welcher Wahrscheinlichkeit ein einzelnes Bit einer Strategie bei der Erzeugung eines Nachkommens kippt.

Im noch überschaubaren Lösungsraum von 256 Strategien scheinen einfache Punktmutationen ausreichend. Zwar wird die sexuelle Reproduktion als eigentlicher Antrieb der Evolution angesehen, doch wird dabei meist auf weitaus größere Lösungsräume Bezug genommen.

Bereits bei Berücksichtigung der eigenen Entscheidungen, bei einer gleichbleibenden Gedächtnistiefe von drei Zügen, würde der Lösungsraum unüberschaubar werden. Dann würde eine Strategie aus $4^3(64)$ Entscheidungen bestehen. Daraus ergeben sich 2^{64} mögliche Strategien, das sind etwa $18 \cdot 10^{18}$ (18 Trillionen) Strategien. Eine entsprechende Simulation wird von Axelrod beschrieben.[Axelrod, 1997]

Zug	1	2	3	4	...
s_{Pawlow}	000/0	001/1	010/1	100/1	000/0 1
$s_{gutmtig}$	1	0	0	0	0
s_{Pawlow}	111/1	111/1 0	111/1	111/1 0	111/1
s_{immerD}	1	1	1	1	1

Tabelle 3.2: Interaktion mit Pawlow (Gedächtnis/nächster Zug)

3.3 Bewertungsfunktion

Die Bewertungsfunktion muss den Erfolg einer Strategie in ihrer Umgebung beschreiben. Dies geschieht durch Bildung eines exponentiell gleitenden Mittelwerts über die erzielten Punkte x aus Interaktionen mit Nachbarn. Dazu wird ein Dämpfungsfaktor q genutzt mit $0 \leq q \leq 1$. Die mittlere Stärke einer Strategie im Zug m lässt sich somit wie folgt ermitteln:

$$strength_{middle} = (1 - q)(x_m + q \cdot x_{m-1} + q^2 \cdot x_{m-2} + \dots) \quad (3.3)$$

$$strength_{middle} = (1 - q) \sum_{i=0}^{\infty} q^i \cdot x_{m-i} \quad (3.4)$$

Mit dem Dämpfungsfaktor q kann die Gewichtung der Vergangenheit bestimmt werden. Während mit einem Dämpfungsfaktor von $q = 0$ der Vergangenheit keine Bedeutung beigemessen wird und nur die letzte Interaktion für die Stärke ausschlaggebend ist ändert sich dies mit steigendem q .

3.4 Auswahloperatoren

Die Auswahloperatoren entscheiden darüber, welches Individuum stirbt und welches Individuum Nachkommen zeugt. In dieser Simulation werden stochastische Auswahlverfahren genutzt. Dazu wird eine Geburtswahrscheinlichkeit b und eine Sterbewahrscheinlichkeit d benötigt. Der Erzeugungsprozess wird mit der Wahrscheinlichkeit b gestartet, der Sterbeprozess mit der Wahrscheinlichkeit d .

Beim Erzeugungsprozess wird ein Nachkomme gemäß den Variationsoperatoren erzeugt. Die Stärke des erzeugenden Individuums wird zwischen dem erzeugenden Individuum selbst und dem neu erstellten Nachkommen aufgeteilt.

Der Sterbeprozess entfernt das Individuum von der Matrix und hinterlässt ein freies Feld.

Um für die Erzeugung von Nachkommen oder den Tod in Frage zu kommen, wird die Stärke eines Individuums betrachtet. Ab einem bestimmten Schwellenwert kann ein Individuum Nachkommen zeugen bzw. sterben.

Es gibt verschiedene Möglichkeiten die Grenzen festzulegen. Eine Möglichkeit ist die Festlegung einer fixen Grenze, welche während der kompletten Simulation konstant bleibt mit $\frac{S+T}{2}$. Mit einer solchen Grenze wird von der Population ein Mindestmaß an Kooperation verlangt. Wird dieses Mindestmaß nicht erreicht, so stirbt die Population aus. Somit ist eine Population von ausschließlich s_{immerD} nicht überlebensfähig.

Ein weiterer Ansatz besteht darin einen globalen Mittelwert über die Stärken der Individuen zu bilden und diesen als Grenze anzusetzen. Hierbei liefert $s(i, j)$ die Stärke des Individuums auf dem $Feld(i, j)$, h ist die Anzahl der in der $n \cdot n$ Matrix vertretenen Individuen.

$$strength_{border} = \frac{1}{h} \sum_{i=0}^n \sum_{j=0}^n s(i, j) \quad h \in N^* \quad (3.5)$$

Als Alternative dazu kann die lokale Mittelwertbildung genutzt werden. Diese erstreckt sich nur über die Nachbarschaft. Hierfür spricht, dass ein Individuum ebenfalls nur mit seiner Nachbarschaft interagiert und auch nur mit dieser in direkter Konkurrenz steht.

In die Simulation werden die globale Mittelwertbildung sowie die lokale Mittelwertbildung aufgenommen, die wahlweise eingesetzt werden können.

Es genügt bereits, wenn Individuen auf der Grenze liegen um für den Erzeugungsprozeß und den Sterbeprozeß in Frage zu kommen. Dadurch bleibt die Evolution in einer homogenen Umgebung nicht stehen.

3.5 Randbedingungen

In das Spiel werden Interaktionsfehler eingebaut, die mit der Wahrscheinlichkeit f auftreten. Jeder Zug eines Spielers wird mit der Wahrscheinlichkeit f umgekehrt. Will ein Spieler kooperieren, so lässt ihn ein auftretender Interaktionsfehler defektieren.

Wenn Strategien auf unbekannte Interaktionspartner treffen, muss ihre Historie vorbelegt werden. Dazu wird ein Parameter r genutzt. Der Parameter r gibt die Wahrscheinlichkeit an, mit der in der Historie eines Individuums eine 1 vorbelegt

wird. Ist $r = 0$ starten alle Interaktionen mit der Historie „000“, bei $r = 1$ mit der Historie „111“

Für die Vorbelegung der Welt muss noch festgelegt werden auf welche Felder ein Individuum gesetzt wird. Dies geschieht mit dem Parameter a , der die Wahrscheinlichkeit für die Belegung eines Feldes angibt.

Ein weiterer Parameter as bestimmt, mit welcher Wahrscheinlichkeit zum Start eine 1 in jeder Strategieposition eines Individuums hinterlegt wird. Während $as = 1$ eine Welt voller s_{immerD} erzeugt, bringt $as = 0$ eine Welt mit s_{immerK} hervor.

4 Die Umsetzung

Das beschriebene Modell ist in großen Teilen bereits in der Simulation von Grams in der Programmiersprache Java umgesetzt [Grams, 2007]. In diesem Abschnitt werden veränderte Funktionalitäten beschrieben sowie die Steuerung der Simulation.

4.1 Auswahl des Interaktionspartners

Für die Auswahl des Interaktionspartners wird ein zufälliger Nachbar aus der Moore-Nachbarschaft bzw. aus der erweiterten Moore-Nachbarschaft ausgewählt.

In Abbildung 4.1 ist eine erweiterte Moore-Nachbarschaft mit $k = 2$ zu sehen.

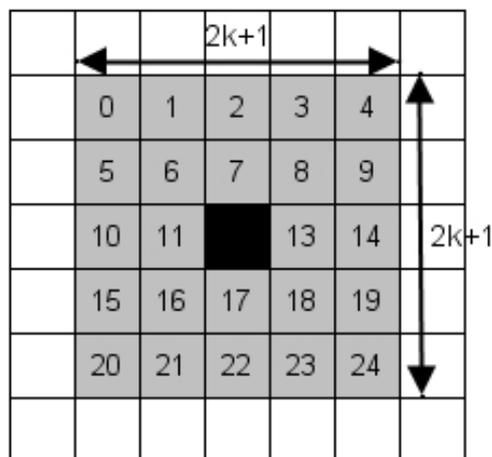


Abbildung 4.1: Anzahl an Nachbarn in einer k -Nachbarschaft

Die Anzahl der Nachbarn N ist somit:

$$N = (2 \cdot k + 1)^2 \quad (4.1)$$

In dieser Nachbarschaft ist auch das Zentrum enthalten, welches beim Finden eines Interaktionspartners ausgelassen werden muss.

Zunächst wird bei der Initialisierung die Nachbarschaftsgröße N ermittelt:

```
1 N=(1+2*k)*(1+2*k);
```

Bei jedem Spielzug muss ein Feld und ein zugehöriges Nachbarfeld ausgewählt werden. Das erste Feld wird durch Ziehen von zwei ganzzahligen Zufallszahlen kleiner n bestimmt. Die beiden Zufallszahlen geben die Koordinaten des Feldes (x_0, y_0) an.

Die Nachbarn werden, wie in Abbildung 4.1 zu sehen, durchnummeriert. Die Auswahl erfolgt durch das Ziehen einer ganzzahligen Zufallszahl i kleiner N . Die relative Position zum Feld (x_0, y_0) kann wie folgt bestimmt werden:

$$\Delta x = i \% (2 \cdot k + 1) - k \quad (4.2)$$

$$\Delta y = \lfloor \frac{i}{(2 \cdot k + 1)} \rfloor - k \quad (4.3)$$

Als letztes muss beachtet werden, dass falls die Mitte getroffen wird die Ziehung wiederholt werden muss.

```
1 int x0 = rand.nextInt(n), y0 = rand.nextInt(n),
2   x1=x0, y1=y0, neighbour;
3   while (x1==x0&&y1==y0) {
4       neighbour=rand.nextInt(N);
5       x1=x0 + ((neighbour%(2*k+1))-k);
6       y1=y0 + ((int)(neighbour/(2*k+1))-k);
7   }
```

4.2 Die Punkt-Mutation

In dem Modell sollen, wie unter Variationsoperatoren auf S.5 beschrieben, statistisch unabhängige Kopierfehler mit der Wahrscheinlichkeit p je Bit auftreten.

Die Hammingdistanzen der Mutationen sind dabei diskret binomialverteilt. Mit der Wahrscheinlichkeit p kippt jedes einzelne Bit einer n bit langen Repräsentation. q beschreibt die Wahrscheinlichkeit, dass ein Bit nicht kippt mit $q = 1 - p$. Daraus ergeben sich die Wahrscheinlichkeitsfunktion $f(x)$ für das Erreichen einer Lösung mit dem Hammingabstand x durch eine Mutation, sowie die resultierende Verteilungsfunktion $F(x)$:

$$f(x) = P(X = x) = \frac{n!}{x!(n-x)!} p^x \cdot q^{n-x} \quad x = 0, 1, 2, \dots, n \quad (4.4)$$

$$F(x) = P(X \leq x) = \sum_{k \leq x} \binom{n}{k} p^k \cdot q^{n-k} \quad (4.5)$$

Unter Verwendung dieser Beziehungen lässt sich die Anzahl der Mutationen eines Nachkommens sehr effizient bestimmen.¹

Es wird ein Array der Länge $n + 1$ angelegt:

```
1 static double[] binomial = new double[strategyLength+1];
```

In dieses Array werden zur Initialisierung die Werte der Verteilungsfunktion $F(x)$ gespeichert:

```
1 for (int i=0; i<=strategyLength; i++){
2     binomial[i]=nOverK(strategyLength, i)
3     *Math.pow(p, i)
4     *Math.pow((1-p), strategyLength-i); //f(x)
5     if (i>0) binomial[i]+=binomial[i-1]; //F(x)
6 }
7 binomial[strategyLength]=1;
```

Um die Anzahl an Mutationen bei einer Reproduktion zu ermitteln wird lediglich eine Zufallszahl zwischen 0 und 1 gezogen und bestimmt, in welches Intervall sie fällt. Die Nummer des Intervalls (beginnend bei 0) ist gleichzeitig die Anzahl der Mutationen.

¹Auf Anregung von Prof. Dr. Timm Grams

```

1 int i=0;
2 double z = rand.nextDouble();
3 while(z>binomial[i])i++; // i= Anzahl Mutationen

```

4.3 Die lokale und globale Grenze

In der Simulation werden variable Grenzen für die Auswahl von Individuen verwendet, die für den Reproduktionsprozess bzw. den Sterbeprozess in Betracht kommen.

Zum einen kann eine globale Grenze verwendet werden. Dabei wird der Mittelwert über die Stärke aller Individuen der Population gebildet. Um die mittlere Stärke der Population effizient zu ermitteln wird die Stärke aller Individuen aufaddiert und bei Änderungen der Stärke um den jeweiligen Wert korrigiert. Die aufaddierte Stärke muss für die Ermittlung der mittleren Stärke der Population durch die Populationsgröße dividiert werden.

Zum anderen kann die Grenze jedoch auch lokal gewählt werden. Bei der lokalen Grenze wird der Mittelwert über die Stärke aller Nachbarn gebildet:

```

1 static double localMeanStrength(int x0, int y0){
2     double res = 0;
3     int x, y, hab=0;
4     for(int i=-k; i<=k; i++)
5         for(int j=-k; j<=k; j++){
6             x=x0+i; y=y0+j;
7             if (x<0)x+=n; else if(n<=x)x-=n; //torusförmig
8             if (y<0)y+=n; else if(n<=y)y-=n;
9             if (world[x][y]!=null){res +=world[x][y].meanStrength(); hab++;}
10        }
11    if (world[x0][y0]!=null){res -=world[x0][y0].meanStrength(); hab--;}
12    if (hab>0)res=res/hab; else res=0;
13    return res;
14 }

```

Die Funktion *localMeanStrength(x₀, y₀)* liefert die mittlere Stärke der Nachbarn des Feldes (x₀, y₀).

4.4 Weitere Änderungen

In der Simulation sind weitere Änderungen vorgenommen. Zur Initialisierung des Spielfeldes wird der Parameter *as* eingeführt. Dieser Parameter legt fest mit welcher Wahrscheinlichkeit eine 1 als Element einer Strategie für die Startbelegung ausgewählt wird.

Die Funktion *initStrategy* liefert Strategien für die Initialisierung der Spielmatrix in Abhängigkeit des Parameters *as*.

```

1  static int initStrategy(){
2      int a=0;
3      for(int i=0; i<8 ;i++){
4          a=a<<1;
5          if(rand.nextFloat(<as) a++;
6      }
7      return a;
8  }
```

Mit der Simulation kann die Historie eines Simulationslaufs aufgezeichnet werden. Dazu wird ein Array verwendet, das ein Feld je Strategie bereitstellt. In jedem Feld wird die aktuelle Anzahl der Individuen mit der entsprechenden Strategie gespeichert.

```

1  static int developing[] = new int[1<<strategyLength];
```

Nach Ablauf jeder Runde wird der Inhalt des Arrays in die Textdatei „Developing.txt“ geschrieben. Zusätzlich wird für jede Spielrunde die mittlere Stärke der Population als Indikator für die Kooperationsbereitschaft hinzugefügt. Dies geschieht durch Aufruf der Funktion *printDeveloping*:

```

1  static void printDeveloping(){ //+
2      String s = rounds + "\t";
3      for(int i=0; i<(1<<strategyLength);i++){
4          s+= developing[i] + "\t";
5      }
6      s+= Double.toString(globalMeanStrength()).replace('.', ',') + "\t";
```

```

7   IO.appendToFile("Developing.txt", s);
8   }

```

Mit einem Tabellenkalkulationsprogramm lässt sich die Datei „Developing.txt“ importieren und nach Wunschkriterien filtern. Auf diese Weise können flexibel Histogramme der Simulationsläufe erstellt werden.

Um einen besseren Überblick bei Simulationsläufen behalten zu können, wird in der Fußleiste des Steuerpults die aktuelle Rundenzahl und die mittlere Stärke der Population eingeblendet.

Für eine einfachere Steuerung ist die Rundenanzahl voreinstellbar. Mit dem Parameter g kann die gewünschte Rundenanzahl festgelegt werden. Nach g Zügen hält die Simulation automatisch an. Bei Bedarf kann die Simulation im Anschluss jedoch weitergeführt werden.

4.5 Die Steuerung

Zur Steuerung der Simulation wird nach dem Starten der Simulation ein Steuerpult geöffnet. Der Button „Eingabe“ lädt die Parameter des vorhergegangenen Simulationslaufs und ermöglicht die Änderung dieser.

In Tabelle 4.1 sind alle Parameter mit ihrer Funktion aufgelistet.

n	Größe der Spielfeldmatrix ($n \cdot n$)
h	Spielzüge je Runde
k	Nachbarschaftsgröße
a	Wahrscheinlichkeit für Startbelegung eines Feldes
as	Wahrscheinlichkeit für „1“ in einer Strategie
b	Geburtswahrscheinlichkeit
d	Sterbewahrscheinlichkeit
p	Mutationswahrscheinlichkeit je Bit
q	Dämpfungsfaktor
r	Vorbelegungswahrscheinlichkeit einer „1“ im Gedächtnis
l	Auswahl der Bewertungsfunktion (1=lokal, 0=global)
kk, kd, dk, dd	Gewinnmatrix (T/R/P/S)
g	Zu simulierende Runden

Tabelle 4.1: Eingabeparameter der Simulation

Nach Festlegung der Parameter wird der Button „Start“ betätigt, der die Initialisierung des Spielfeldes zur Folge hat.

Mit dem Button „Step“ kann die Simulation rundenweise bedient werden. Der Button „Run“ lässt die Simulation fortlaufend starten. Mit „Break“ wird die Simulation wieder angehalten.

Der Button „Ende“ schließt die Simulation. Durch Betätigen des Buttons „Hilfe“ können Informationen über die Steuerung des Programms und die Parameter abgerufen werden.

In der Spielmatrix werden die Individuen mit unterschiedlichen Farben dargestellt. Schwarze Felder sind leer. Die Farben der Individuen entsprechen ihren Basischarakteren. In Tabelle 4.2 ist die jeweilige Farbe der Basischaraktere ersichtlich. Je größer die Hammingdistanz einer Strategie zu ihrem Basischarakter, desto heller wird sie dargestellt.

Tit for Tat	(255, 0, 0)	rot
Vergelter	(255, 0, 255)	magenta
Immer K	(255, 255, 0)	gelb
Pawlow	(0, 0, 255)	blau
Immer D	(0, 255, 0)	grün
Hinterlist	(0, 255, 255)	cyan

Tabelle 4.2: Farbcodierung der Basischaraktere in dreidimensionalen Farbwerten (Rot, Grün, Blau) [Grams, 2007]

Ein Mausklick in der Spielmatrix auf ein Individuum zeigt die Wesen im Detail. In der Tabelle 4.3 sind die Informationen dazu näher erläutert.

Identität	Identifizierungsnummer und Position des Wesens
Lebenskraft	gleitender Mittelwert über Interaktionen (siehe S.17)
Basistyp	nächster Basistyp (siehe Tabelle 3.1)
Entfernung	Hammingdistanz vom Basistyp
Strategie	Darstellung der Strategie (siehe Abbildung 3.3)
Erfahrungen	letzte Interaktionen mit Nachbarn

Tabelle 4.3: Informationen zu Wesen

In der Fußleiste des Steuerpults werden die Anzahl der Spielzüge und die mittlere Stärke der Population angezeigt.

Durch Aktivieren der Controllbox „Historie aufzeichnen“ wird die Historie des Simulationsverlaufs in der Datei „Developing.txt“ abgelegt. Es muss vor dem Start der Simulation festgelegt werden, ob die Historie aufgezeichnet werden soll, ein nachträgliches Aktivieren der Aufzeichnung ist nicht möglich.

5 Versuche

5.1 Die Entstehung der Kooperation

Um die Entstehung von Kooperation zu untersuchen, muss von einem Zustand der Defektion ausgegangen werden. Thomas Hobbes (1588-1679) beschreibt einen Naturzustand ohne zentrale Autorität durch Staat und Gesetze. In einem solchem Zustand gilt das Naturrecht in dem jeder Mensch gegen jeden Krieg führt.

Einen Zustand ohne zentrale Autorität ist auch im vorliegenden Simulationsmodell zu finden. Die Individuen in der Simulation können keine Verträge eingehen und das Verhalten anderer erzwingen. Um das unkooperative Verhalten nachzubilden, wird die Vorbelegungswahrscheinlichkeit für eine 1 der Strategien $as = 1$ gesetzt. Damit wird das Spielfeld nach der Initialisierung ausschließlich mit den Strategien s_{immerD} vorbelegt.

Die Strategie s_{immerD} ist, wie schon von Axelrod gezeigt wurde, eine evolutionär stabile Strategie. Eine evolutionär stabile Strategie kann nicht von einem einzelnen Individuum mit einer anderen Strategie invadiert werden. Um eine Strategie zu invadieren, muss der Invasor mit der vorherrschenden Strategie eine höhere Punktezahl erzielen als die vorherrschende Strategie untereinander. Dies ist bei der Strategie s_{immerD} nicht möglich, da ein Gegner gegen diese Strategie in keinem Fall über die Auszahlung P (Punishment for Defection) hinauskommt, welche die Strategie untereinander erhält.[Axelrod, 2006]

Kommen die Invasoren jedoch in Gruppen und nicht einzeln nacheinander, so kann eine evolutionär stabile Strategie verdrängt werden. Axelrod untersucht den Fall, dass eine homogene evolutionär stabile Population s_{immerD} von einer Gruppe Invasoren angegriffen wird. Die Zahl der Invasoren ist im Verhältnis zu den Individuen der vorherrschenden Strategie vernachlässigbar. Die erzielte Punktezahl der vorherrschenden Population ändert sich somit nicht.

Würden die Interaktionspartner der Invasoren zufällig ausgewählt werden, so würden sie fast immer auf Individuen der vorherrschenden Population treffen und diese dadurch nicht invadieren können. Treffen sie jedoch zu einem Prozentsatz v auf andere Invasoren können sie durch gute Ergebnisse untereinander mehr Punkte

als die vorherrschende Strategie erzielen. Die durchschnittlichen Punkte x die ein Invasor I erhält setzen sich somit aus den durchschnittlichen Punktzahlen zusammen, die er aus der Interaktion mit einem anderen Invasor ($A(I|I)$) und einem Individuum der vorherrschenden Population ($A(I|P)$) erhält:

$$x = v \cdot A(I|I) + (1 - v) \cdot A(I|P) \quad (5.1)$$

Strategien die eine Population von s_{immerD} auf diese Weise durchdringen können sind nette, vergeltende Strategien wie $s_{Vergelter}$ oder $s_{TitForTat}$.

In der Simulation von Lomborg kommen Invasoren nicht einzeln nacheinander, sondern in größeren Gruppen. Auch bei zufälliger Wahl der Interaktionspartner in der gesamten Population kann unter den von Lomborg beschriebenen Bedingungen kooperatives Verhalten entstehen.[Lomborg, 1996]

In der hier umgesetzten Simulation kommen Invasoren jedoch nacheinander an, was die asynchrone Entwicklung der Individuen bedingt. Durch die Punktmutation und die Reproduktion können kleinste lokale Gruppen von Invasoren entstehen. Interagiert diese Gruppe mit zufälligen Individuen der gesamten Population kann sie sich nicht durchsetzen.

Der Mechanismus der Territorialität sorgt dafür, dass ein Invasor bei kleiner k -Nachbarschaft (z. B. $k = 1$) zu einem bestimmten Teil auf andere Invasoren trifft. Die Territorialität hat jedoch auch zur Folge, dass die vorherrschende Population in den Randgebieten mit den Invasoren zu einem nicht zu vernachlässigenden Anteil interagiert.

Interagiert beispielsweise s_{immerD} mit $s_{Vergelter}$ erhält s_{immerD} im ersten Zug T und anschließend P , $s_{Vergelter}$ erhält im ersten Zug S und anschließend ebenfalls P . In den Interaktionen untereinander erhält $s_{Vergelter}$ konstant R (wenn keine Interaktionsfehler auftreten). Um den anfänglichen Nachteil in der Kennenlernphase mit s_{immerD} ausgleichen zu können, muss $s_{Vergelter}$ lange genug existieren. Deshalb muss die Sterberate d ausreichend niedrig sein. Ist dies der Fall, kann $s_{Vergelter}$ die vorherrschende Strategie durchdringen.

Mit der Simulation kann betrachtet werden, wie sich eine Population entwickelt, wenn die Territorialität zum Tragen kommt in einer k -Nachbarschaft mit $k = 1$ und was passiert, wenn die Interaktionspartner aus der gesamten Spielmatrix ausgewählt werden.

Dazu wird die Größe der Spielmatrix auf $n = 41$ festgelegt. Die k -Nachbarschaft wird mit $k = 20$ bzw. mit $k = 1$ definiert. Bei $k = 20$ kann ein Individuum auf der Matrix mit allen anderen Individuen interagieren. Die weiteren Parameter sind der Tabelle 5.1 zu entnehmen.

Größe der Spielfeldmatrix ($n \cdot n$)	41
Spielzüge je Runde h	100.000
Nachbarschaftsgröße k	20 bzw. 1
Wahrscheinlichkeit für Startbelegung eines Feldes a	0,5
Wahrscheinlichkeit für „1“ in einer Strategie as	1
Geburtswahrscheinlichkeit b	0,01
Sterbewahrscheinlichkeit d	0,005
Mutationswahrscheinlichkeit je Bit p	0,001
Wahrscheinlichkeit für Interaktionsfehler f	0,001
Dämpfungsfaktor q	0,9
Vorbelegungswahrscheinlichkeit einer „1“ im Gedächtnis r	0
Bewertungsfunktion	lokal
Gewinnmatrix (T/R/P/S)	4/2/0/-1

Tabelle 5.1: Parameterwahl zum Vergleich von lokaler und globaler Grenze

Nach der Initialisierung sieht das Spielfeld wie in Bild 1 von Abbildung 5.1 aus. Es ist eine Population von s_{immerD} zu sehen.

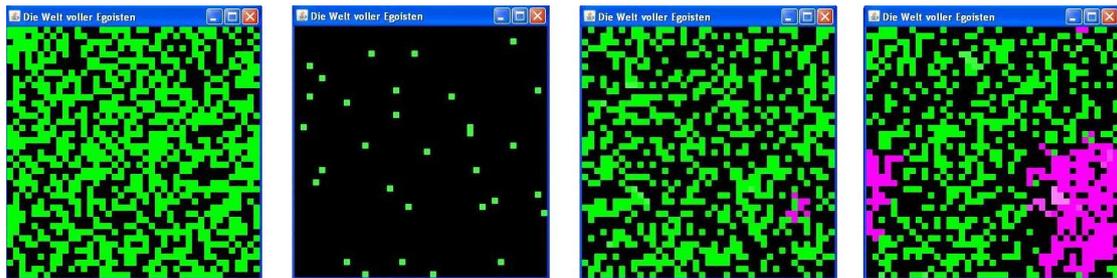


Abbildung 5.1: Vergleich lokal und global, Bild 1: Startbelegung, Bild 2: global nach 587 Spielrunden, Bild 3: lokal nach 22 Spielrunden, Bild 4: lokal nach 60 Spielrunden

In dem globalen Versuchslauf mit $k = 20$ dünnt sich die Populationsdichte sehr schnell aus. Selbst nach über 500 Runden kann sich keine nette Strategie in dieser Welt behaupten. Vereinzelt tauchen nette Strategien auf, die jedoch schnell wieder aussterben. Dieser Zustand ist in Bild 2 von Abbildung 5.1 zu sehen.

Der lokale Versuchslauf mit $k = 1$ zeigt bereits nach 22 Spielrunden eine kleine Gruppe von Vergeltern (siehe Bild 3, Abbildung 5.1). Diese Gruppe setzt sich nach und nach gegen die umliegenden defektierenden Strategien durch und wächst weiter. Nach 60 Spielrunden stellt sie bereits einen beachtlichen Teil der Population

dar (siehe Bild 4, Abbildung 5.1). Nach 136 Runden ist die Strategie *simmerD* vollständig ausgestorben.

Die Simulation zeigt, dass die Territorialität die Entstehung der Kooperation in einer Welt voller defektierender Individuen begünstigt. Trotz der anfänglich geringeren Auszahlung gegenüber der vorherrschenden Strategie *simmerD* gelingt es den netten, vergeltenden Strategien sich durchzusetzen, indem sie untereinander kooperieren.

5.2 Die gutmütigen Strategien

Die in der Simulation stattfindenden Interaktionen sind fehlerbehaftet. Mit der Wahrscheinlichkeit f treten Interaktionsfehler auf. Diese Interaktionsfehler haben einen großen Einfluss auf das Maß an Kooperation. Interaktionsfehler können die Kooperationsbereitschaft zwischen zwei netten Strategien einbrechen lassen.

In Tabelle 5.2 sind zwei Beispiele zu sehen. Eine Interaktion zwischen zwei Vergeltern und eine Interaktion zwischen zwei Tit for Tat Strategien. Bei Interaktionen mit Vergeltern reicht bereits ein Kooperationsfehler und die Vergelter stellen ihre Kooperationsbereitschaft mit dem entsprechenden Partner dauerhaft ein. In der Tabelle ist zu sehen, dass bei der Interaktion der Vergelter im 2. Zug ein Interaktionsfehler auftritt. Daraufhin stellt das betrogene Individuum die Kooperationsbereitschaft ein. In allen folgenden Begegnungen defektieren die beiden Vergelter.

Der in der Simulation nachgebildete Vergelter würde nach dreimaligem Kooperieren des Partners ebenfalls wieder kooperieren. Dazu müssten allerdings drei Interaktionsfehler in Folge auftreten.

Bei der Strategie TFT hat ein einzelner Interaktionsfehler noch nicht zur Folge, dass die Kooperationsbereitschaft vollständig eingestellt wird. Ein einzelner Interaktionsfehler zwischen zwei Individuen mit der Strategie TFT führt zu wechselseitiger Ausnutzung. In Tabelle 5.2, Zug 2 tritt ein Interaktionsfehler auf. In den folgenden Zügen ist die gegenseitige Ausnutzung zu beobachten, bis ein weiterer Interaktionsfehler auftritt. Der folgende Interaktionsfehler kann die Kooperation wieder herstellen oder, wie in Tabelle 5.2 Zug 7 gezeigt, die Kooperation vollständig zum Erliegen bringen.

Auf Interaktionsfehler kann auf verschiedene Art und Weise reagiert werden. Mit Reue und mit Großmut[Axelrod, 2006]. Kann ein Spieler feststellen, dass ein Interaktionsfehler bei der Ausführung seiner Aktion aufgetreten ist, hat er die Mög-

Zug	1	2	3	4	5	6	7	8	...
$s_{Vergeltter}$	0	$\bar{1}$	0	1	1	1	1	1	...
$S_{Vergeltter}$	0	0	1	1	1	$\bar{0}$	1	1	...
s_{TFT}	0	$\bar{1}$	0	1	0	1	$\bar{1}$	1	...
S_{TFT}	0	0	1	0	1	0	1	1	...

Tabelle 5.2: Fehlerhafte Interaktion: Interaktionsfehler sind mit einem Balken gekennzeichnet ($\bar{1}$)

lichkeit Reue zu zeigen. Dies geschieht dadurch, dass er über eine Bestrafung die seinem Interaktionsfehler folgt hinwegsieht.

Es ist jedoch nicht davon auszugehen, dass den Akteuren Informationen über Interaktionsfehler immer zur Verfügung stehen. In dieser Simulation haben die Individuen keine Informationen über auftretende Interaktionsfehler. Somit kann Reue in dieser Simulation nicht auftreten.

Großzügigkeit ist eine weitere Alternative auf Interaktionsfehler zu reagieren. Großzügigkeit bewirkt, dass auf einzelne Defektionen des Gegners nicht mit Vergeltung reagiert wird.

Um die Reaktion auf Interaktionsfehler zu testen, werden Simulationsläufe mit den Parametern aus Tabelle 5.3 durchgeführt, wobei die Interaktionsfehlerwahrscheinlichkeit f verändert wird.

Größe der Spielfeldmatrix ($n \cdot n$)	100
Spielzüge je Runde h	1.000.000
Nachbarschaftsgröße k	2
Wahrscheinlichkeit für Startbelegung eines Feldes a	0,2
Wahrscheinlichkeit für „1“ in einer Strategie as	1
Geburtswahrscheinlichkeit b	0,01
Sterbewahrscheinlichkeit d	0,005
Mutationswahrscheinlichkeit je Bit p	0,001
Dämpfungsfaktor q	0,9
Vorbelegungswahrscheinlichkeit einer „1“ im Gedächtnis r	0
Bewertungsfunktion	lokal
Gewinnmatrix (T/R/P/S)	4/2/0/-1

Tabelle 5.3: Parameterwahl zur Untersuchung der Reaktion auf Interaktionsfehler

Mit $f = 0$ treten keine Interaktionsfehler auf. In diesem Fall verdrängt eine nette vergeltende Strategie die vorherrschende Strategie s_{immerD} . Dies ist aufgrund des

geringen Hammingabstandes meist die Strategie $s_{Vergelter}$.

Da keine Interaktionsfehler auftreten, kann keine andere Strategie bessere Ergebnisse erzielen als eine nette, vergeltende Strategie (wie $s_{Vergelter}$). Das hat zur Folge, dass die vorherrschende Strategie dominant bleibt. Es treten lediglich neutrale Mutationen auf, deren Häufigkeit von der Mutationsrate p abhängt. Neutrale Mutationen sind Mutationen, die dem Individuum weder Vor- noch Nachteile einbringen [Motoo, 1980]. In einer Umgebung von ausschließlich netten, vergeltenden Strategien ohne Interaktionsfehler ist jede Mutation neutral solange die Strategie nett bleibt.

Mit einer Interaktionsfehlerwahrscheinlichkeit von $f = 0,01$ wird die Strategie s_{immerD} wesentlich langsamer durch nette, vergeltende Strategien verdrängt.

Parallel dazu entstehen defektierende Strategien, die sich wechselseitig ausnutzen können, wie in Tabelle 5.4 zu sehen am Beispiel der Strategie $s_{11110111}$. Diese Art der wechselseitigen Ausnutzung wird durch Interaktionsfehler hervorgerufen. Dieser Mechanismus erscheint nicht sehr effektiv, gibt den Teilnehmern jedoch bereits einen Vorteil gegenüber der Strategie s_{immerD} .

Zug	1	2	3	4	5	6	7	8	9	...
$s_{11110111}$	1	1	0	1	1	1	1	1	0	...
$s_{11110111}$	1	1	1	1	1	0	1	1	1	...

Tabelle 5.4: Wechselseitige Ausnutzung zweier Strategien, hervorgerufen durch Interaktionsfehler (auftretende Interaktionsfehler sind mit einem Balken gekennzeichnet ($\bar{1}$))

Erst wenn die nicht netten Strategien durch konsequent vergeltende Strategien verdrängt wurden, können sich gutmütige Strategien in der Population etablieren. Dabei treten häufig gutmütige Varianten der Strategie s_{TFT} auf wie in Tabelle 5.5 zu sehen. Diese Strategien verzichten je nach Situation auf eine Bestrafung, so kann eine Spirale von gegenseitiger Defektion verhindert werden.

Original Tit for Tat	10 10 10 10
Variante 1	10 10 10 00
Variante 2	10 10 00 10
Variante 3	10 00 10 10

Tabelle 5.5: Gutmütige Varianten von Tit for Tat

Die Variante 1 lässt sich beispielsweise den ersten Betrug eines Gegners gefallen und hält die Kooperation aufrecht. Bei nochmaligem Betrug innerhalb der nächsten

zwei Züge wird jedoch Vergeltung geübt.

Bei gutmütigen Strategien besteht jedoch die Gefahr, dass sie ausgenutzt werden. So kommt es vor, dass in der Nachbarschaft von gutmütigen Gruppen, die sich an der Strategie s_{immerK} orientieren defektierende Strategien entstehen und damit die zu großmütigen Strategien dezimieren.

5.3 Das Quasi-Spezies-Modell

Manfred Eigen beschäftigt sich mit der Frage wie das Neue entsteht und sich etabliert. Dabei spielen die Faktoren Variabilität und Stabilität eine zentrale Bedeutung. Damit Neues entstehen kann wird zum einen Variabilität gebraucht. Jedoch ist auch eine gewisse Ordnung nötig damit nicht Chaos herrscht. Diese Ordnung verlangt Stabilität.

In der Spannung zwischen Freiheit und (selbstgewählten) Spielregeln liegt der Schlüssel, wie stark wir das Neue hineinlassen wollen.

[von Pierer, Oetinger, 1997, S. 17]

Das Quasi-Spezies-Modell ist ein mathematisches Modell, das hierzu interessante Ansätze liefert. Es werden Populationen von Individuen betrachtet. Die Individuen können durch eine Bitfolge der Länge n repräsentiert werden. Die Umwelt wird als konstant angenommen. In dieser Umwelt hat jedes Individuum abhängig von der Repräsentation eine bestimmte Lebenserwartung und Chance auf Nachkommen. Somit besteht für jedes Muster i eine Geburtenrate (Replication Rate) A_i und eine Sterberate (Degredation Rate) D_i .

Die Replikation der Muster ist fehlerbehaftet. Mit der Kopierfehlerwahrscheinlichkeit p wird ein einzelnes Bit falsch übertragen. q ist die Kopiertreue mit $q = 1 - p$. $d(i, j)$ bezeichnet die Hammingdistanz zweier Muster. Mit der Wahrscheinlichkeit Q_{ij} erzeugt das Muster j einen Nachkommen mit dem Muster i mit $Q_{ij} = p^{d(i,j)} \cdot q^{n-d(i,j)}$. Die Wachstumsrate $W_{i,j}$ ist $W_{ii} = Q_{ii} \cdot A_i - D_i$ für $j = i$ und $W_{ij} = Q_{ij} \cdot A_j$ für $j \neq i$.

Das Wachstums eines Muster hängt somit von allen Populationsgrößen N der Muster ab. Die Wachstumsgeschwindigkeit einer Population ist damit $\dot{N}_i = \sum_j W_{ij} \cdot N_j$.¹

¹Näheres zum Quasi-Spezies-Modell sowie ein Beispiel in Form einer Tabellenkalkulation sind auf der Website von Grams erhältlich (Stand 14.04.2008): <http://www2.hsfulda.de/grams/OekoSim1/QuasiSpezies.htm>

Je größer die Kopierfehlerwahrscheinlichkeit p , desto früher wird der stationäre Endzustand erreicht. Im stationären Endzustand existiert das stärkste Muster („Stammsequenz“) mit den anderen Mutantensequenzen. Die Verteilung wird als Quasi-Spezies bezeichnet.

Was sich bereits bei einfachen Beispielen mit $n = 3$ zeigt ist noch deutlicher bei längeren Bitmustern zu beobachten. Ab einer bestimmten Kopierfehlerwahrscheinlichkeit p setzt sich die Stammsequenz nicht mehr von den anderen Mutantensequenzen ab und geht im „Chaos“ unter. Bei längeren Bitmustern wird der Übergangsbereich noch deutlicher. [Grams, 2008]

Für die Simulation werden die Parameter wie in Tabelle 5.6 gewählt, wobei verschiedene Simulationsläufe mit ändernder Mutationswahrscheinlichkeit p ausgeführt werden.

Größe der Spielfeldmatrix $(n \cdot n)n$	100
Spielzüge je Runde h	1.000.000
Nachbarschaftsgröße k	2
Wahrscheinlichkeit für Startbelegung eines Feldes a	0,02
Wahrscheinlichkeit für „1“ in einer Strategie as	1
Geburtswahrscheinlichkeit b	0,02
Sterbewahrscheinlichkeit d	0,01
Wahrscheinlichkeit für Interaktionsfehler f	0,001
Dämpfungsfaktor q	0,9
Vorbelegungswahrscheinlichkeit einer „1“ im Gedächtnis r	0
Bewertungsfunktion	lokal
Gewinnmatrix (T/R/P/S)	4/2/0/-1

Tabelle 5.6: Parameterwahl zur Untersuchung der Empfindlichkeit auf die Mutationswahrscheinlichkeit p

Bei geringer Mutationswahrscheinlichkeit mit $p = 0,001$ verläuft die Evolution nur langsam. In Abbildung 5.2 ist ein Histogramm eines Simulationslaufs mit den zehn stärksten Strategien der ersten 1000 Runden zu sehen. In Abbildung 5.3 links ist der Zustand der Spielfeldmatrix nach 1000 Runden zu sehen.

Zunächst wird in der Anfangsphase die vorherrschende Strategie s_{immerD} von einer netten, vergeltenden Strategie verdrängt. Daraufhin können sich gutmütigere Strategien in der Umgebung behaupten.

Nach der Anfangsphase geht die Population in einen relativ stabilen Zustand über. Die Kooperationsbereitschaft bleibt weitgehend auf gleichem Niveau, was durch eine konstante mittlere Stärke der Population zum Ausdruck kommt.

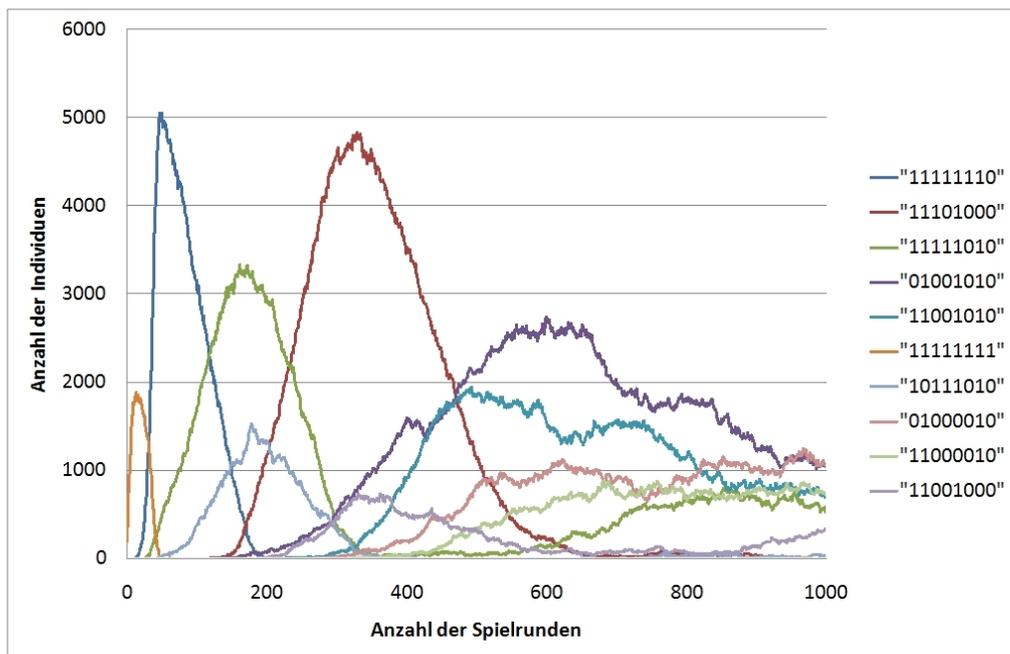


Abbildung 5.2: Histogramm eines Simulationslaufs mit der Mutationswahrscheinlichkeit $p=0,001$

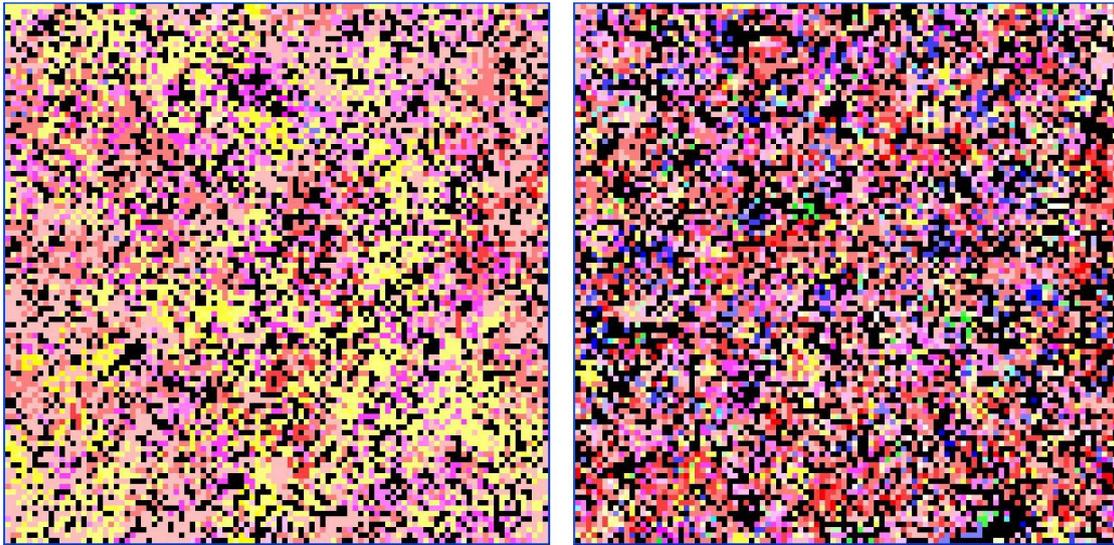


Abbildung 5.3: Spielfeldmatrix, links: nach 1000 Runden mit $p=0,001$, rechts: nach 800 Runden mit $p=0,05$

Auf der Spielmatrix bilden sich relativ große verbundene Gruppen von gleichartigen Individuen aus. In der Nähe von Gruppen mit sehr gutmütigen Strategien entstehen zeitweise neue Strategien des Basistyps *simmerD*. Diese verdrängen die ausnutzbaren Strategien, können sich jedoch nicht gegen andere weniger großzügige Strategien behaupten.

Mit steigender Mutationswahrscheinlichkeit p wird, wie im Quasi-Spezies-Modell, die Evolution beschleunigt und die Anfangsphase verkürzt sich. Die Gruppengrößen von gleichartigen Individuen verkleinern sich und die Artenvielfalt nimmt zu.

In der Simulation lässt sich eine so deutliche Grenze wie in dem Quasi Spezies Modell nicht feststellen. Die Übergänge von stabilen zu instabilen Zuständen sind fließend.

Bei einer Mutationswahrscheinlichkeit von $p = 0,05$ wird bereits keine Stabilität mehr erreicht. Die mit geringerer Mutationswahrscheinlichkeit zu beobachtende Anfangsphase ist nur noch ansatzweise in den ersten Zügen zu erkennen. Die vertretenen Strategien unterliegen starken Schwankungen, wie das Histogramm in Abbildung 5.4 zeigt, das die zehn Strategien mit den im Verlauf meisten Anhängern abbildet. Es stellt sich kein stabiler Zustand mehr ein. Keiner Strategie gelingt es über längere Zeit dominant zu sein und sich abzusetzen.

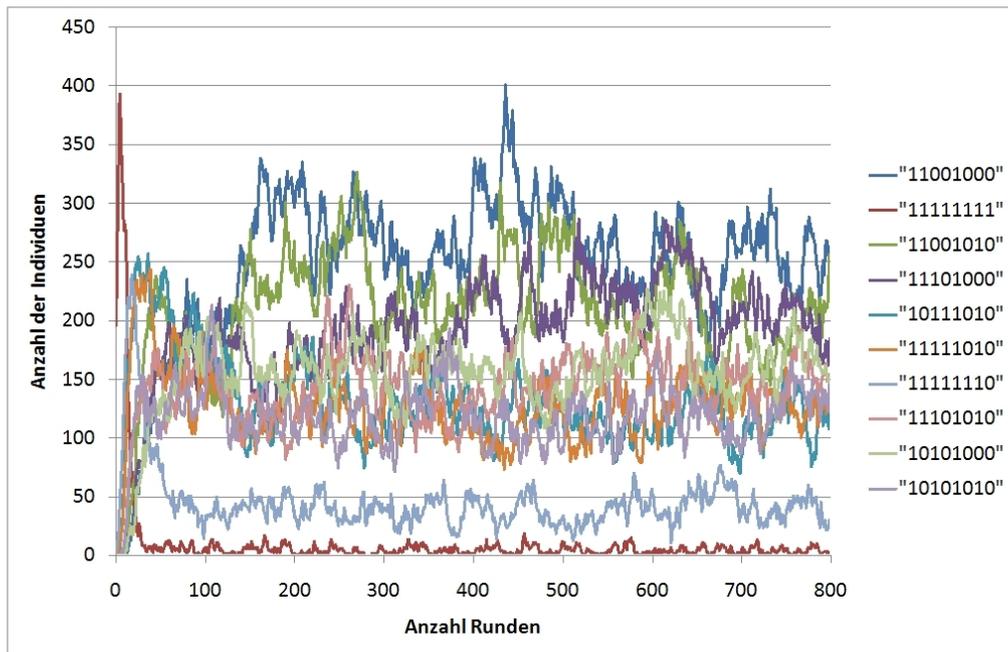


Abbildung 5.4: Histogramm eines Simulationslaufs mit der Mutationswahrscheinlichkeit $p=0,05$

Die Strategien mit den meisten Anhängern aus Abbildung 5.4 stellen nach 800 Runden nur einen kleinen Teil von 22,11% der gesamten Population, was auf die große Artenvielfalt schließen lässt. Im Simulationslauf mit geringer Mutationswahrscheinlichkeit mit $p = 0,001$ stellen die abgebildeten zehn Strategien nach 800 Runden 73,86% der Population dar. Das zeigt, dass es diesem Zustand einige wenige dominante Strategien gibt.

6 Schlussbetrachtung

Mit der erstellten Simulation ist es gelungen aus einem Zustand in welchem nur Egoisten die Welt bevölkern in einen Zustand mit kooperierenden Individuen zu gelangen.

Die Simulation zeigt, dass unter Verwendung der Territorialität auch nacheinander eintreffende Invasoren eine evolutionär stabile Strategie von s_{immerD} durchdringen können. Voraussetzung dafür ist eine genügend hohe Lebenserwartung der Invasoren.

Die Territorialität sorgt dafür, dass die von Axelrod beschriebene nötige Wahrscheinlichkeit für das Aufeinandertreffen zweier Invasoren vorhanden ist. Ohne den Mechanismus der Territorialität gelingt die Entstehung von kooperativen Verhalten in diesem Modell nicht.

Das Ergebnis steht im Einklang mit Beobachtungen in der Natur. Revierbildende männliche Vögel können andere Artgenossen an ihrem Gesang identifizieren. Hören sie einen unbekanntes männlichen Artgenossen sind die Tiere um vieles aggressiver, als wenn sie einen Nachbarn aus ihrer eigenen Umgebung hören [Wilson, 1975]. In lokalen Umgebungen, in denen gleiche Akteure häufig aufeinandertreffen kann Kooperation leichter entstehen als in globalen Umgebungen, in denen sich gleiche Akteure nur selten begegnen.

Als Reaktion auf Interaktionsfehler entstehen in der Simulation gutmütige Strategien, die häufig nah mit der Strategie Tit For Tat verwandt sind. Gutmütigkeit ist der einzige Mechanismus, welcher mit den in der Simulation abgebildeten Strategien, umgesetzt werden kann. Aufgrund der beschränkten Informationen, welche den Individuen zur Verfügung stehen, sind sie nicht in der Lage mit einem anderen Mechanismus wie Reue auf Interaktionsfehler zu reagieren.

Die Simulation mit dem einfachen zugrunde liegenden Modell kann keine realen Situationen nachbilden. Reale Situationen sind zu komplex, Entscheidungen beruhen auf einer Vielzahl von Faktoren und sind mitunter zufallsbedingt. Was allerdings mit diesem Modell gelingt, ist ein Grundverständnis für die Entstehung von Kooperation bei Interaktionen zu vermitteln.

Durch die wenigen Prämissen können die Ergebnisse auf weite Bereiche übertragen werden. Das reicht von der Ausbildung sozialer Kontakte bis hin zur Entstehung von Kooperation auf Zellebene.

Mit der erstellten Simulation können weitere Betrachtungen und Versuche durchgeführt werden. Interessante Ergebnisse verspricht aber auch die Modifikation der Simulationen. So können Auswirkungen von anderen Variationsoperatoren betrachtet werden oder mit Änderung der Repräsentation der Individuen die Ausbildung komplexerer Strategien ermöglicht werden.

Literaturverzeichnis

- [Axelrod, 2006] Axelrod, Robert: The Evolution of Cooperation. Revised Edition, Basic Books, New York 2006
- [Axelrod, 1997] Axelrod, Robert: The Complexity of Cooperation - Agent Based Models of Competition and Collaboration. University Press, New Jersey 1997
- [Grams, 2007] Grams, Timm: Evolution kooperativen Verhaltens. <http://www2.hs-fulda.de/grams/OekoSimSpiele/KoopEgoProgramm/-KoopEgo.pdf>(Stand: 05.03.2008)
- [Grams, 2008] Grams, Timm: Umweltsimulation mit Tabellenkalkulation: Quasi-Spezies: Stabilität und Variabilität in der Evolution. <http://www2.hs-fulda.de/grams/OekoSim1/QuasiSpezies.htm>(Stand: 14.04.2008)
- [Holland, 1992] Holland, John H.: Genetische Algorithmen. Spektrum d. Wiss.(1992) 9, 44-51
- [Lomborg, 1996] Lomborg, Bjørn: Nucleus and Shield: The Evolution of Social Structure in the Iterated Prisoner's Dilemma. American Sociological Review, Vol 61, No.2 2.(Apr., 1996), 278-307
- [Mehlmann, 1997] Mehlmann, Alexander: Wer gewinnt das Spiel?: Spieltheorie in Fabeln und Paradoxa. Vieweg, Braunschweig/Wiesbaden 1997
- [Michalewicz/Fogel, 2000] Michalewicz, Zbigniew; Fogel, David, B.: How to Solve It: Modern Heuristics. Springer-Verlag, Berlin, Heidelberg 2000
- [Motoo, 1980] Motoo, Kimura: Die „neutrale“ Theorie der molekularen Evolution. Spektrum d. Wiss.(1980) 1, 94-102
- [Nowak, May und Sigmund, 1995] Nowak, Martin A.; May Robert M.; Sigmund Karl: Das Einmaleins des Miteinander. Spektrum d. Wiss.(1995) 8, 46-53
- [Gerhardt/Schuster, 1995] Gerhardt Martin; Schuster Heike: Das digitale Universum: Zelluläre Automaten als Modelle der Natur. Vieweg, Braunschweig/Wiesbaden 1995
- [von Pierer, Oetinger, 1997] Pierer, Heinrich von; Oetinger, Bolko von: Wie kommt das Neue in die Welt? Carl Hanser Verlag, Wien 1997

[Wilson, 1975] Wilson, Edward Osborne: Sociobiology. The New Synthesis. Belknap Press, Cambridge, Massachusetts, 1975, 2000

Eidesstattliche Erklärung

Ich versichere hiermit, dass ich meine Diplomarbeit mit dem Thema

Simulierte Kreativität am Beispiel der Entstehung kooperativen Verhaltens

selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Fulda, den 19. Juni 2008

HENRIK SCHAAKE

A Anhang

A.1 Inhalt CD

Auf der beiliegenden CD sind die vorliegende Diplomarbeit im pdf-Format sowie die Software und Quelltexte zur erstellten Simulation enthalten.

Inhalt der CD	Inhalt.txt
Diplomarbeit im pdf-Format	Diplomarbeit.pdf
Software mit Quelltexten	KoopEgo