

Selbst-Organisation und Evolution mit offenem Ausgang

Dissertation zur Erlangung des Doktorgrades
des Fachbereichs Naturwissenschaften I - Physik
der Bergischen Universität Gesamthochschule
Wuppertal

vorgelegt von
Harald Freund
aus Wuppertal

WUB-DIS 92-7
August 1992

Abstract

The bit string model is a new kind of evolutionary model. In contrast to systems known from the literature it contains already two hierarchical levels. This way a structure is introduced in state space, without which it seems hard to speak about evolution or complexity. Although an open evolution was one of the final aims (which could not be achieved) there are a number of interesting results.

The rules valid in the model are as simple as possible to avoid that complex behavior is built-in. One important result is the evidence that the use of startcodons in the decoding scheme is hard to avoid. This seems to have biological relevance since startcodons are used in natural decoding processes. Another interesting result concerns the age distribution of species. Although no really open-ended evolutionary path could be found, we already see an age distribution that can best be described by a power law. This hints at a system with a wide range of effective time scales.

During this research random models were a valuable tool. They help to find out which results are due to evolutionary dynamics and which have already been built into the rules. They also allow to find analytic formulas for, e.g., age distributions, which oddly enough usually are not looked at in other evolutionary models.

In the third appendix a new type of walk with memory is studied. This might serve as a toy model for the behavior one must adopt to avoid exhaustion of resources and attraction of parasites and predators. The walk takes place on a regular square lattice with periodic boundary conditions. Although the walk is completely deterministic, it mimics a “true” self-avoiding walk, i.e. a random walk with weak autocorrelation. This shows that the red queen effect can lead to aperiodic behavior. In addition to the case of single walkers in a flat landscape we also study the cases of hilly landscapes and of several walkers performing simultaneous walks.

Inhalt

Abbildungsverzeichnis	v
Tabellenverzeichnis	vii
1 Einleitung	1
2 Artificial Life	3
2.1 Hyperzyklen und katalytische Netzwerke	4
2.2 Hügelige und flache Fitnesslandschaften	5
2.3 Weitere Modelle	7
3 Das Bit-String-Modell	8
3.1 Die Motivation des Modells	8
3.2 Die Hierarchie	9
3.3 Die Dekodierung	10
3.4 Der verwendete genetische Algorithmus	11
3.4.1 Die Wechselwirkungsphase	11
3.4.2 Die Selektion	13
3.4.3 Die genetischen Operatoren	15
3.5 Die Meßgrößen	16
3.6 Das Energieflußmodell	18
4 Die Bedeutung von Zufall	19
4.1 Unterschiedliche Aspekte des Zufalls	19
4.2 Alternative Implementationen	20
4.3 Das Zufallsmodell	22
5 Simulationen des Bit-String-Modells	23
5.1 Verschiedene Leseschemata	23
5.2 Typische Ergebnisse des Zufallsmodells	23
5.3 Höchste Packungsdichte	25
5.3.1 Ergebnisse von Simulationen	25
5.3.2 Erklärung der Ergebnisse	25
5.3.3 Der Einfluß der Selektion	28
5.3.4 Geringere Korrelationen	29
5.4 Das Dekodierschema <i>Gen an Gen</i>	30
5.4.1 Die Dekodierung	30
5.4.2 Der Einfluß von Punktmutationen	30
5.4.3 Simulationsergebnisse	31
5.4.4 Bewertung des Dekodierschemas <i>Gen an Gen</i>	33

5.5	Verwendung von Startcodons	33
5.5.1	Die Dekodierung	33
5.5.2	Der Aufbau von Startcodons	34
5.5.3	Erste Ergebnisse	34
5.5.4	Die Informationslänge	35
5.5.5	Variation der Individuenlänge	37
5.5.6	Abhängigkeit von den Anfangsbedingungen	38
5.5.7	Speziesspuren und evolutionäre Aktivität	39
5.6	Die Bedeutung von Spezies	41
5.7	Variationen des Modells	43
5.8	Zusammenfassung	45
6	Das Evolving-Bugs-Modell	47
6.1	Einfache Version	47
6.2	Kolonien bildende Bugs	49
6.3	Evolutionäre Aktivität	52
6.3.1	Strategische Bugs	52
6.3.2	Aktivitätswellen	53
6.3.3	Auswertung von Simulationen	55
6.3.4	Nachweis von Evolution bei den strategischen Bugs	60
7	Emergenz	63
7.1	Zwei Definitionen von NAGEL	63
7.2	Die Definition von LANGTON	64
7.3	Die Definition von ASSAD und PACKARD	65
7.4	Zusammenfassung	65
8	Zusammenfassung	66
A	Die Altersverteilung der Individuen	69
A.1	Konstante Eliminationswahrscheinlichkeit	69
A.2	Altersabhängige Eliminationswahrscheinlichkeit	70
B	Verzweigungsprozesse	72
B.1	Kurzer historischer Überblick	72
B.2	Der Galton-Watson-Prozeß	72
B.3	Anwendung auf das Bit-String-Modell	74

C Der Wanderer im Sumpf	77
C.1 Motivation	77
C.2 Die Metapher der <i>Roten Königin</i>	77
C.3 Ein einzelner Wanderer im Sumpf	78
C.3.1 Beschreibung des Modells	78
C.3.2 Erste Konsequenzen	79
C.3.3 Erste Simulationsergebnisse	80
C.3.4 Die statistischen Eigenschaften des Modells	82
C.4 Erweiterungen und Verallgemeinerungen	84
C.4.1 Dynamischer Zufall	84
C.4.2 Unendliche Gitter	85
C.4.3 Langsame Relaxation	86
C.4.4 Hügelige Landschaften	88
C.4.5 Mehrere Wanderer	90
C.5 Diskussion des Modells	92
 Literaturverzeichnis	 94

Abbildungsverzeichnis

2.1 Katalytischer Hyperzyklus	4
3.1 Genotyp-Phänotyp-Relation	9
3.2 Der Genbaum zu Tabelle 3.1	11
3.3 Crossover	16
3.4 Cut-and-Splice	16
5.1 Sterbewahrscheinlichkeiten im Zufallsmodell	24
5.2 Genbesetzungszahlen im Zufallsmodell	24
5.3 Genbesetzungszahlen bei höchster Packungsdichte	25
5.4 Individuum mit höchster Packungsdichte	26
5.5 Vergleich verschiedener Wechselwirkungsraten	28
5.6 Genbesetzungszahlen bei niedrigerer Packungsdichte	29
5.7 Dekodierung ohne Überlapp	30
5.8 Herausschneiden des mutierten Gens	31
5.9 Beispiel für <i>genetic takeover</i>	31
5.10 Beispiel für Bistabilität	32
5.11 Individuum mit Startcodons	33
5.12 Genbesetzungszahlen bei Verwendung von Startcodons	35
5.13 Zeitliche Entwicklung der Informationslänge	36
5.14 Informationsgehalt bei verschieden langen Bit-Strings	36

5.15	Zeitliche Entwicklung der Bewertung	37
5.16	Entwicklung der Bitlänge bei Verwendung von Cut-and-Splice	37
5.17	Länge der Anfangspopulation 20 Bits	38
5.18	Länge der Anfangspopulation 60 Bits	39
5.19	Speziesspuren	40
5.20	Speziesspuren im Zufallsmodell	41
5.21	Vergleich Individuen- und Speziesaltersverteilung	41
5.22	Vergleich Individuen- und Speziesaltersverteilung	42
5.23	Speziesaltersverteilung im Vergleich mit dem Zufallsmodell	43
5.24	Vergleich Sortiermodell und Turniermodell	44
5.25	Speziesspuren für eine sehr große Population	45
6.1	Aktivitätswellen im Evolving-Bugs-Modell	54
6.2	Altersverteilung der Bugs	55
6.3	Dynamik des Futterfeldes	56
6.4	Vergleich der Bevölkerungsgrößen	57
6.5	Aktivitätswellen bei gradient ascent	58
6.6	Aktivitätswellen bei zufälliger Bewegung	59
6.7	Dynamik des Futterfeldes bei gradient ascent	59
6.8	Dynamik des Futterfeldes bei zufälliger Bewegung	60
6.9	Die bevorzugten Bewegungsrichtungen	61
A.1	Eliminationswahrscheinlichkeiten im Bugsmodell	71
B.1	Vergleich Galton-Watson-Prozeß und Daten	75
C.1	Die Wege des Wanderers	82
C.2	Der Weg des Wanderers	83
C.3	Die regelabhängige Einschwingphase	84
C.4	Die Verteilung $P(l)$	85
C.5	Das mittlere Abstandsquadrat $\langle R_n^2 \rangle / n$	85
C.6	Die Autokorrelation c_n	86
C.7	Die Verteilung der Wiederkehrzeiten	86
C.8	Das mittlere Abstandsquadrat auf unendlichen Gittern	87
C.9	Die Abstandsverteilung ohne Relaxation	88
C.10	Die Varianz der Tiefenverteilung	88
C.11	Der Weg in einer hügeligen Landschaft	89
C.12	Flucht aus einem Maximum	90
C.13	Die Abstandskorrelation für mehrere Wanderer	91
C.14	Wahrscheinlichkeit für Rückschritte	92

Tabellenverzeichnis

3.1	Die Bitdarstellung des Genbaums	10
3.2	Eine verwendete Wechselwirkungsmatrix $A(g_i, g_k)$	12
5.1	Eine zweite Wechselwirkungsmatrix $A(g, g')$	26
5.2	Teilsequenz mit höchster Packungsdichte	26
5.3	Überlappende Gengruppen	27
5.4	Vergleich von Einzel- und Gruppenwechselwirkungen	27
6.1	Beispiel für ein Artorg-Gehirn	50
C.1	Transientenzeiten bis zum Erreichen eines periodischen Orbits	81

1 Einleitung

Bei dem in dieser Arbeit untersuchten Thema, *Selbst-Organisation und Evolution mit offenem Ausgang*, handelt es sich um ein interdisziplinäres Thema, das international von einer kleinen, aber sehr aktiven Gemeinschaft von Forschern betrieben wird, deren wissenschaftliche Herkunft äußerst vielfältig ist. Neben theoretischen Physikern sind vor allem Informatiker, Biologen, Chemiker und Mathematiker beteiligt. Einen guten Überblick über den Stand der Forschung auf diesem Gebiet geben die von LANGTON *et al.* herausgegebenen Proceedings zweier Konferenzen über *Artificial Life* [82, 84]. Die vielfältige Herkunft macht es möglich, daß die untersuchten Probleme aus ganz unterschiedlichen Richtungen angegangen werden, da jeder der Forscher sein Grundwissen und seine eigenen Arbeitsmethoden mitbringt. Dies ist einer der Gründe dafür, daß ich mich als theoretischer Physiker mit dem doch so *unphysikalischen* Thema, *Evolution und Leben*, auseinandergesetzt habe.

In der Elementarteilchenphysik ist es heute fast nur noch dann möglich, Fortschritte zu erzielen, wenn man Forschung in Großprojekten betreibt, wie dies in Europa am CERN und am DESY geschieht. Dort forschen hunderte bis tausende von Physikern in großen Kollaborationen. In der theoretischen Physik ist dieser Trend zu großen Arbeitsgruppen noch nicht so deutlich. Es gibt zwar theoretische Physiker, die eng mit den Experimentatoren der oben genannten Großprojekte zusammenarbeiten, und es existieren auch relativ kleine Gruppen bei Berechnungen in der Gittereichtheorie, diese benötigen allerdings riesige Mengen von Rechenzeit auf Großrechnern, seien dies Vektorrechner wie die Cray, oder Parallelrechner wie die Transputer oder die Connection Machine.

Normalerweise aber arbeitet ein theoretischer Physiker in sehr kleinen Arbeitsgruppen. Dies gilt auch für Forscher, die sich mit Artificial Life beschäftigen. Dort ist es noch möglich, mit geringem Aufwand sehr interessante Ergebnisse zu erhalten. Diese Arbeit z.B. wurde zu großen Teilen auf Workstations der Firma SUN durchgeführt. Das Modell, das hier behandelt wird, kann mit einer seriellen Rechnerarchitektur sehr gut bearbeitet werden. Bevor man Rechenzeit auf so teuren Maschinen wie der CM2 verschwendet, sollte man lieber zunächst das Problem (den Forschungszweig) mit konventionellen Mittel ausloten, was bei dieser Arbeit gemacht wurde. Dabei stellt sich heraus, daß es möglich ist, Abwandlungen des Modells zu erfinden, die gut zu der Architektur z.B. einer Connection Machine passen. Da die Arbeit zu dem Zeitpunkt aber schon weit fortgeschritten war, wurde dies dann nicht mehr als zwingend notwendige Erweiterung im Rahmen dieser Arbeit angesehen.

Für einen theoretischen Physiker ist das Forschungsgebiet Artificial Life deshalb interessant, weil man sich mit dem Phänomen *Leben* als Folge der Selbst-Organisation der Materie auseinandersetzt. Es ist mittlerweile wohlbekannt, daß dabei bereits bei einfachsten Systemen chaotisches Verhalten entstehen kann. Dies suggeriert, daß auch komplexe Formen der Organisation bereits bei der Verwendung von einfachen Regeln entstehen können. In der Natur beobachtet man, daß zunächst die Chemie und später auch die Biologie immer komplexer geworden sind. Eine allgemein anerkannte Definition des Begriffs *Komplexität* wird sowohl in der Mathematik als auch in der Physik seit Jahrzehnten vergeblich gesucht. Man hat noch kein Maß definieren können, das die intuitive Beurteilung in jeder Beziehung widerspiegelt.

Die mathematische Komplexitätstheorie befaßt sich mit der Komplexität von Algorithmen. Ein Algorithmus ist umso komplexer, je mehr Zeit er verbraucht, um sein Ergebnis zu erzielen (der Speicherplatzverbrauch kann ebenfalls ein Kriterium sein). Zum Beispiel

ist die Suche nach dem Maximum von N Zahlen sehr einfach, sie verbraucht nämlich nur $N - 1$ Vergleiche bzw. Rechenoperationen. Das Sortieren der N Zahlen [76] ist komplexer, weil es mehr Rechenzeit verbraucht. Die Suche nach dem kürzesten Weg durch N Städte (das ist das *Travelling Salesman Problem*) gehört zu der Klasse der vermutlich komplexesten Probleme (siehe z.B. [46]). In der Physik und Biologie untersucht man weniger Algorithmen, sondern eher Muster, die man ebenfalls als einfach, komplex oder zufällig einstufen möchte. KOLMOGOROV [78] hat als Maß die Länge des kürzesten Programms vorgeschlagen, das dieses Muster erzeugt. Wenn auf der Menge der Muster ein Wahrscheinlichkeitsmaß definiert ist, dann entspricht der Mittelwert dieser Länge der SHANNON-Information [109] bzw. der Entropie. Diese ist für zufällige Sequenzen maximal und damit ungeeignet zur Definition der Komplexität. Mehrere physikalische Definitionen werden von GRASSBERGER in [51] diskutiert. Damit eine Größe als Maß für Komplexität geeignet ist, muß sie kleine Werte sowohl für geordnete als auch für zufällige Muster annehmen. Die Untersuchungen in Artificial Life können helfen, die Begriffe *Komplexität*, *Selbst-Organisation* und *Leben* besser zu verstehen. Sie können also zu einer anerkannten Definition dieser Begriffe führen.

Artificial Life ist ein Gebiet, das erst seit wenigen Jahren intensiv bearbeitet wird. Da die Forscher aus unterschiedlichen Bereichen kommen, existiert eine Vielzahl von Modellen, einige davon werden in Kapitel 2 kurz vorgestellt. Ein weiteres Modell, das Bit-String-Modell, wird in Kapitel 3 eingeführt. Da der Begriff der Komplexität so diffus ist, ist es auch schwierig, Observablen allgemeiner Art zu finden, die auf die Anwesenheit komplexen Verhaltens schließen lassen könnten. Ein wichtiger Teil der vorliegenden Arbeit ist es daher, mögliche Meßgrößen zu definieren, was zum Teil schon in Kapitel 3 geschieht.

Die Diskussion in den ersten Kapiteln läßt erkennen, daß es den Forschern sehr leicht gemacht wird, ihr eigenes Süppchen in Form eines eigenen Modells zu kochen. Damit wird das große Potential des interdisziplinären Forschungszweiges Artificial Life auch gleich als sein größtes Problem entlarvt. Es existiert kein *Standardmodell*, wie z.B. das Ising-Modell, an dem man Dinge austesten und mit dem man Ergebnisse vergleichen kann. Dies ist der Grund, warum in dieser Arbeit starkes Gewicht auf den Vergleich mit sogenannten *Zufallsmodellen* gelegt wird, die in Kapitel 4 diskutiert werden. Sie ermöglichen es teilweise, analytische Formeln zu erhalten, wie dies in den Anhängen A und B vorgeführt wird. Bei der Diskussion von verschiedenen Varianten des Bit-String-Modells in Kapitel 5 wird davon intensiv Gebrauch gemacht. Die Idee der Zufallsmodelle kann aber nicht nur im Bit-String-Modell verwendet werden, wie die Diskussion in Kapitel 6 zeigt. Dort ist es sogar möglich, falsche Konzepte zu entlarven.

Bei der Diskussion der ganzen Artificial Life Modelle taucht immer wieder der Begriff *Emergenz* auf. Dieser Begriff ist ähnlich schlecht zu definieren wie *Komplexität* oder *Leben*, dies wird bei der Diskussion verschiedener Ansätze in Kapitel 7 deutlich. In Kapitel 8 werden die wichtigsten Ergebnisse dieser Arbeit noch einmal zusammengefaßt. Den Abschluß der Arbeit bildet Anhang C, in dem noch ein weiteres Modell (der Wanderer im Sumpf) vorgestellt wird. Dieses Modell gehört zwar weniger in den Bereich Artificial Life, als mehr in die statistische Physik der Irrwege, dennoch deckt es einen Grundlagenbereich ab, der die Ergebnisse des Bit-String-Modells ergänzt.

2 Artificial Life

Diese Arbeit gehört in das seit wenigen Jahren bestehende Gebiet *Artificial Life*, was *künstliches Leben* bedeutet. Da Artificial Life zu einem feststehenden Begriff geworden ist, wird auf eine Eindeutschung in dieser Arbeit verzichtet.

Während sich die Biologie mit dem Leben befaßt, das wir auf der Erde beobachten, versucht der Forschungszweig Artificial Life, den Begriff in einem weiteren Sinne zu verstehen. In der Biologie ist Leben als Phänomen sehr eng mit dem Substrat verbunden, in dem es gefunden wird, man geht also **nur** von einer Kohlenstoff-Chemie aus. Es besteht aber die Möglichkeit, daß Leben auch in anderer Form existieren kann. Daher sollte ein Forschungszweig, der sich mit dem Phänomen *Leben* auseinandersetzt, nicht nur *das Leben, wie wir es kennen*, sondern auch *das Leben, wie es sein könnte*, untersuchen [83].

In seinem Einleitungsartikel der ersten Proceedings [82] über Artificial Life gibt LANGTON [83] einige Kriterien für Artificial Life Systeme an. Das wichtigste Kriterium ist, daß sie grundsätzlich eine *bottom-up* Struktur aufweisen, d.h., es werden nur die Grundbausteine und ihre Wechselwirkungen untereinander definiert. Globales Verhalten baut darauf auf und ergibt sich erst durch die Dynamik.

Dynamik ist ein weiterer wichtiger Aspekt in diesem Forschungszweig. Während in anderen Bereichen, wie z.B. der Optimierung, das Resultat (z.B. die Gewinnmaximierung) der Zielpunkt ist, ist in Artificial Life die fortdauernde Dynamik das eigentliche Forschungsobjekt. Man versucht mit seinen Modellen, die Eigenschaften und das Verhalten lebender Systeme nachzuzeichnen. Werden dabei die richtigen Regeln verwendet, so erzeugt man ein Modell, dessen äußere Form zwar unterschiedlich sein mag, dessen Verhalten aber das Original kopiert. An dieser Stelle seien nur kurz die sogenannten *Boids* von REYNOLDS [105] erwähnt, die sich wie ein Vogelschwarm verhalten.

LANGTON gibt eine Liste von Eigenschaften an, die computergestützte Artificial Life Modelle erfüllen sollten [83]. Diese wird im Folgenden zitiert:

- Artificial Life Systeme bestehen aus Populationen einfacher Programme oder Spezifikationen.
- Es gibt kein einzelnes Programm, das alle anderen leitet, d.h., es gibt keinen *globalen Controller*.
- Jedes Programm beschreibt (bzw. definiert), wie sich *einfache* Einheiten in ihrer *lokalen* Umgebung verhalten und wie sie mit anderen einfachen Einheiten wechselwirken.
- Es gibt keine Regeln, außer den physikalischen Gesetzen, die *globales* Verhalten beschreiben.
- Jedes Verhalten auf einer Ebene, die höher als die individuellen Programme ist, ist daher *emergent*. (Welche Schwierigkeiten es bereitet, den Begriff *Emergenz* sauber zu definieren, wird anhand der Diskussion in Kapitel 7 klar werden.)

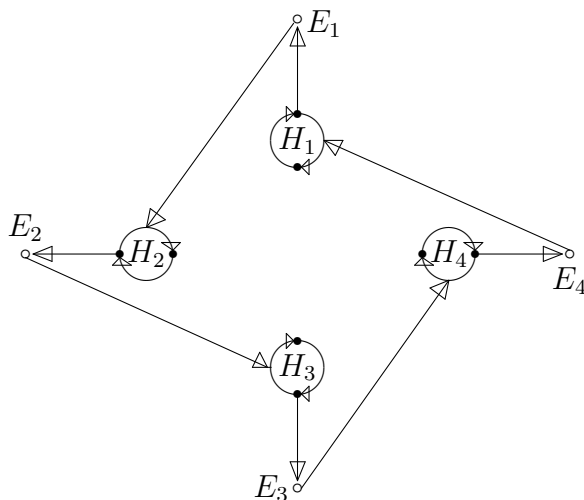
Diese Forderungen stecken den Rahmen ab. Es existieren aber viele Möglichkeiten, diesen mit *Leben* zu erfüllen, weil das Leben selbst in so vielen Erscheinungsformen vorliegt. Einige der in der Literatur untersuchten Modelle werden in den folgenden Abschnitten kurz vorgestellt.

2.1 Hyperzyklen und katalytische Netzwerke

Die Erbinformation (d.h. Genome) von selbst einfachen Lebewesen bestehen schon aus mindestens 10^4 Nukleotiden [73]. Wenn die Information linear vorwärts dekodiert wird, wie es in der DNS der Fall zu sein scheint, dann kann bei einer Replikation schon eine niedrige Fehlerrate die Zerstörung der gesamten Information bewirken. Bei einer *normalen* Fehlerrate von 10^{-3} bis 10^{-2} können damit nur Genome bis zu einer Länge von etwa 10 bis 100 Nukleotiden stabil bleiben [90]. Für größere Genome sind Fehlerkorrekturmechanismen nötig. Es ist bekannt, daß *höhere* Lebensformen unterschiedliche Mechanismen besitzen [80]. Diese Mechanismen haben aber Genome, die wiederum mindestens aus 1 000 bis 10 000 Nukleotiden bestehen. Es stellt sich also die Frage: Wie sind solche Korrekturmechanismen überhaupt entstanden, wenn man voraussetzt, daß alles Leben aus der *einfachen* Ursuppe der Erde von vor etwa 4 Milliarden Jahren entstanden ist? EIGEN und SCHUSTER entwickelten dafür das Konzept der *Hyperzyklen* [30], das zumindest theoretisch das Problem der *Fehlerkatastrophe* löst.

Ein katalytischer Zyklus ist definiert als ein geschlossener Ring $\{n_1, n_2, \dots, n_k\}$, von $k \geq 1$ sich gegenseitig katalysierenden Elementen. Ein einfaches Beispiel für einen solchen Zyklus H_1 mit zwei Elementen stellt der RNA-Replikationsvorgang dar, da dieser über die Zwischenstufe eines Komplements abläuft.

Abbildung 2.1: Katalytischer Hyperzyklus



Diese Abbildung zeigt einen katalytischen Hyperzyklus, der aus 4 Einzelzyklen mit jeweils zwei Elementen besteht. Ein solches Gebilde ist stabiler gegen den Ausfall einer Komponente, wenn mehrere Elemente in der Lage sind, eines der nachfolgenden Elemente im Hyperzyklus zu katalysieren. (Entnommen aus [119])

Wird in einem solchen Zyklus H_i ein weiteres Produkt E_i erzeugt, das als Katalysator für ein Element eines anderen Zyklus H_{i+1} dient, so läßt sich daraus ein Hyperzyklus, also ein *Zyklus von Zyklen* konstruieren. Abbildung 2.1 zeigt als Beispiel einen Hyperzyklus, der aus vier Einzelzyklen mit jeweils zwei Elementen besteht [119].

Ein solcher Hyperzyklus kann wesentlich mehr Information speichern als ein einzelner Zyklus, für den auch weiterhin die Beschränkung auf etwa 100 Nukleotide bestehen bleibt.

Experimentelle Beweise für die Richtigkeit der Hyperzyklushypothese sind bisher nicht gelungen. MAYNARD-SMITH argumentiert sogar, daß Hyperzyklen sehr instabil sind, weil sie anfällig gegen die Invasion von Parasiten sind [90]. Ein Parasit in diesem Zusammenhang ist ein Element, daß von Mitgliedern des Hyperzyklus katalysiert wird, aber selber *keines* der Mitglieder katalysiert. Diese Überlegungen basieren darauf, daß Hyperzyklen üblicherweise mit Differentialgleichungen beschrieben werden. Dies entspricht einer Welt ohne Fluktuationen. Wie aber unter anderem die Untersuchungen in dieser Arbeit zeigen, sind gerade Fluktuationen ein sehr wichtiger Bestandteil evolutionärer Systeme. BOERLI-

JST und HOGEWEG [15, 16] haben das Hyperzyklusmodell als zellularen Automaten auf einem Gitter implementiert, wobei statt Differentialgleichungen probabilistische Regeln verwendet werden. Sie zeigen in dieser Variation des Hyperzyklusmodells, daß Parasiten auf lange Sicht in den meisten Fällen eliminiert werden. WOLTER bestätigt diese Ergebnisse und zeigt, daß die Ergebnisse auch stabil bei Parametervariationen sind [120].

Mit einem ähnlichen Ziel sind auch die Untersuchungen zu autokatalytischen Netzwerken begonnen worden [6, 7, 33, 68]. Sie stellen den Versuch dar, aus kurzen Polymeren über Katalyse einzelne längere Polymere mit bedeutenden Konzentrationen zu erzeugen, was durch spontane Synthese nicht möglich ist. In Erweiterungen dieser Studien [8, 9] werden die so geformten autokatalytischen Netzwerke als ein einzelner Organismus aufgefaßt, der selber wieder der Evolution unterworfen ist. Bei diesen Studien wird meistens zufällig entschieden, welche Polymere welche Reaktion katalysieren.

FONTANA verwendet bei der Entwicklung seiner *Algorithmic Chemistry* [37] einen anderen Ansatz: Ausgehend vom λ -Calculus der mathematischen Logik [23] baut er seine eigene Chemie auf. Seine Individuen sind Programme bzw. genauer gesagt Funktionen, die die anderen Individuen als Argument auffassen und daher auswerten können. Bei der Simulation einer Population zufällig erzeugter Funktionen entstehen normalerweise (d.h., wenn man alle *Reaktionen* zuläßt) sogenannte Kopierer, also Funktionen, die andere und sich selbst kopieren. Sie beherrschen sehr schnell die Dynamik. Sobald man die Regeln soweit ändert, daß die Kopierreaktion als nicht zulässige Reaktion verworfen wird, bilden sich autokatalytische Netzwerke aus [37]. Obwohl FONTANA ein sehr interessantes Modell behandelt, hat es vom Standpunkt dieser Arbeit aus einen bedeutenden Nachteil: Es existieren keine Mutationen bzw. Kopierfehler. Im Zusammenhang der Algorithmic Chemistry bedeutet das: Alle Reaktionen laufen vollständig deterministisch ab. Die einzige Stelle, an der Zufall ins Spiel kommt, ist die zufällige Anfangspopulation und die Entscheidung, welche Funktion welche andere momentan als Argument bearbeitet.

Nachdem in diesem Abschnitt Modelle vorgestellt wurden, die sehr stark an der *wirklichen Chemie* orientiert sind, werden im nächsten Abschnitt Modelle behandelt, die, wie es auch in dieser Arbeit geschieht, Individuen als Bit-Strings kodieren.

2.2 Hügelige und flache Fitnesslandschaften

KAUFFMAN und JOHNSEN [71] untersuchen in ihrem Modell den Aspekt der *Co-Evolution*. Sie behandeln eine Klasse von Modellen (die NK-Modelle [69]), die auf einer *hügeligen* Fitnesslandschaft leben. Ihre Individuen sind Bit-Strings. Jedes der N Bits wird als ein Gen aufgefaßt, das in zwei Allelen vorliegen kann. Der Fitnessbeitrag jedes der N Gene hängt *epistatisch* (zu deutsch in etwa: gleichmäßig bzw. gleich stark) von K anderen Genen im selben Individuum ab. Bei genetischen Veränderungen wird nur ein Gen *umgeklappt*. Für den Fall $K = 0$ hängt die Bewertung des Gens nur von seiner eigenen Einstellung ab. Jedes Gen kann also unabhängig von den anderen Genen seinen *optimalen* Wert annehmen. Dies entspricht einem einzigen Maximum der Fitnesslandschaft. Wird die Anzahl K der epistatischen Wechselwirkungen erhöht, so wird die Fitnesslandschaft hügeliger. Ein einzelnes Gen kann nun nicht mehr unabhängig von den anderen mit ihm verbundenen Genen seinen eigenen Wert so einstellen, daß sein Fitnessbeitrag optimal wird. Für den Fall $K = N - 1$ wird die Landschaft schließlich völlig unkorreliert.

Bei der einfachen Version des NK-Modells [69] versuchen alle Individuen durch Mutationen das Optimum (die Optima) der Fitnesslandschaft aufzusuchen. Eine Erweiterung sieht vor,

daß S verschiedene Spezies existieren und die Bewertung eines Gens nun nicht nur von den K anderen Genen im selben Individuum (derselben Spezies) abhängt, sondern noch zusätzlich von C Genen in jeder (oder zumindest einigen) der S anderen Spezies. Auf diese Weise werden die Fitnesslandschaften der einzelnen Spezies miteinander gekoppelt.

Der Begriff Co-Evolution wird in diesem Zusammenhang verwendet, da die genetische Veränderung einer Spezies die Fitnesslandschaften der anderen Spezies deformiert. Die Dynamik des Systems hängt sehr stark von dem Grad der Kopplung der Landschaften ab. Sehr häufig bilden sich sogenannte *Nash-Gleichgewichte* [100]. Dies ist ein Begriff aus der Spieltheorie. Auf diesen Kontext übertragen liegt ein Nash-Gleichgewicht vor, wenn für jede Spezies gilt, daß eine genetische Veränderung nicht zu einer Verbesserung führt, solange die anderen Spezies genetisch unverändert bleiben. Es ist aber genauso möglich, daß sich *periodische Orbits* bilden, bei denen jede Spezies eine Folge von *Ausprägungen* durchläuft. Wenn Nash-Gleichgewichte vorliegen, kann eine von außen eingeführte Veränderung einer einzigen Fitnesslandschaft eine *Lawine* von Veränderungen in allen anderen Spezies auslösen. Die Wahrscheinlichkeitsverteilung für die Größe der Lawinen kann dabei durch ein Potenzgesetz beschrieben werden [71]. Dies wird in Verbindung gebracht mit den Massensterben in der Erdhistorie [103] und mit dem Konzept der *selbstorganisierten Kritikalität* in der Physik [10].

Co-Evolution wird von mehreren Autoren untersucht: HILLIS setzt Parasiten ein, um die Optimierungseigenschaften eines Evolutionsalgorithmus zu verbessern [58]. Im Prinzip verwenden alle Modelle, die Parasiten enthalten, Co-Evolution (siehe z.B. [67]). Genauso gilt dies natürlich auch für alle Systeme, bei denen die anderen Mitglieder der Population einen Teil der Umwelt bilden [65, 81].

In diesem Sinne ist auch das in dieser Arbeit beschriebene Bit-String-Modell (siehe Kapitel 3 und 5 und [42, 43]) ein Modell der Co-Evolution, da sich dort ebenfalls durch eine geänderte Zusammensetzung der Bevölkerung die Überlebenschancen einzelner Individuen drastisch ändern können.

Während die hügelige Fitnesslandschaft sehr stark an Optimierungsprobleme erinnert, gehen DERRIDA, HIGGS und PELITI [27, 56, 57] einen anderen Weg. Sie untersuchen das Verhalten einer Population in einer flachen Fitnesslandschaft, in der alle Individuen mit gleicher Wahrscheinlichkeit Nachwuchs zeugen können. Sie schaffen dadurch eine Verbindung zur Spin-Glas-Theorie (einen Überblick gibt [94]), was es ihnen ermöglicht, analytische Ergebnisse herzuleiten. Obwohl die flache Fitnesslandschaft zunächst nur von akademischem Interesse zu sein scheint, ist sie doch ein wertvolles Werkzeug. Es wird nämlich gezeigt [27],

- daß das genaue Wissen über das Verhalten einer Population, die sich ohne Selektion weiterentwickelt, es einfacher macht, herauszufinden, welche Eigenschaften auf den Einfluß der Selektion selbst zurückzuführen sind.
- daß die hergeleiteten Ergebnisse unter gewissen Bedingungen auch Relevanz für Evolutionsmodelle auf einer hügeligen Landschaft besitzen.

Speziell der erste Punkt ist auch ausschlaggebend für die Verwendung von *Zufallsmodellen* als Referenzsystem für das Bit-String-Modell.

In Erweiterungen der Untersuchungen von [27] wird asexuelle durch sexuelle Reproduktion ersetzt [108, 56, 57]. Nun können die aus der statistischen Physik bekannten Rechenmethoden verwendet werden, um Aussagen über Speziesformation und genetische Vielfalt zu machen. Die Untersuchungen von KIMURA auf molekularer Ebene deuten an, daß die

flache Fitnesslandschaft doch nicht so akademisch ist, wie sie zunächst scheint, wenn diese Theorie der *neutralen Selektion* auch sehr umstritten ist [72].

2.3 Weitere Modelle

Die Modelle aus Abschnitt 2.1 sind sehr stark an der wirklichen Chemie orientiert. Die Modelle aus Abschnitt 2.2 sind an der statistischen Physik orientiert und können daher zum Teil analytisch behandelt werden. Im Gegensatz dazu sind eine ganze Reihe von Modellen in der Literatur vorhanden, die ihre eigene Welt definieren, und so versuchen, dem Phänomen *Leben* oder einigen seiner Ausprägungen näher zu kommen (siehe z.B. [101, 102], [66, 81], [85] und [4, 11, 96]).

RASMUSSEN *et al.* [101, 102] haben *Coreworld* entwickelt. Die Welt ist der Speicherplatz eines Computers, die *Lebewesen* sind die Programmzeiger, die sich teilen und sterben können. Die Regeln werden nur auf der untersten Ebene definiert, es sind die Maschinenbefehle. Die *Welt* ist ständigen Veränderungen unterworfen, da die Programmstücke, auf die über die Zeiger zugegriffen wird, sich selbst oder andere Programme verändern können. In diesem Zusammenhang wird der Begriff *programmierbare Materie* verwendet. Die Autoren beobachten unterschiedliche Evolutionspfade in Abhängigkeit von ihren Anfangsbedingungen. Es bilden sich kooperative Strukturen aus, die mit Organismen gleichgesetzt werden.

JEFFERSON *et al.* [66] und KOZA [81] betrachten vollständige Computerprogramme als Individuen, die der Evolution unterworfen werden. Bei der Selektionsphase wird eine Fitness genommen, die angibt, wie gut ein bestimmtes Problem gelöst wird. Damit wird bei diesen Systemen eine globale Fitnessfunktion vorgegeben, die in der Natur **nicht** existiert. Ihre Untersuchungen sind also besonders nützlich bei der Behandlung von Optimierungsproblemen.

LINDGREN [85] untersucht das sogenannte *Iterated Prisoner's Dilemma* (IPD), ein Modell in der Spieltheorie [5, 100]. Bei ihm sind die Individuen Strategien, mit denen man das IPD spielt. Er beobachtet unter anderem *Punctuated Equilibria* (zu deutsch: *durchbrochene Gleichgewichte*), also Phasen der scheinbaren Stasis, die sich mit kurzen Phasen der vollständigen Umstrukturierung des Systems abwechseln. Bei der Umstrukturierung finden Massensterben statt, die vom dynamischen System selbst ausgelöst werden und die daher keine externen Katastrophen brauchen [1, 103].

Als letztes Modell sei noch auf das Evolving-Bugs-Modell von PACKARD *et al.* hingewiesen [4, 11, 96], das ausführlich in Kapitel 6 diskutiert wird.

Während sich bei den Modellen aus Abschnitt 2.2 die zu verwendenden Meßgrößen aus der Analogie zu den Systemen der statistischen Physik ergeben, ist speziell bei den Modellen dieses Abschnitts nicht klar, welche Meßgrößen genommen werden müssen, um die interessanten Phänomene zu finden. Damit stellt in diesen Modellen, wie auch im Bit-String-Modell, die Datenanalyse einen nicht-trivialen Aspekt der Arbeit dar.

3 Das Bit-String-Modell

In diesem Kapitel wird das Bit-String-Modell so allgemein wie möglich vorgestellt. Dies läßt noch viel Raum für Variationen, deren Sinn oder Unsinn erst durch Simulationen geklärt werden kann. Deshalb werden im Kapitel 5 ausführlich Simulationen dargestellt, die auf den Konzepten dieses Kapitels aufbauen.

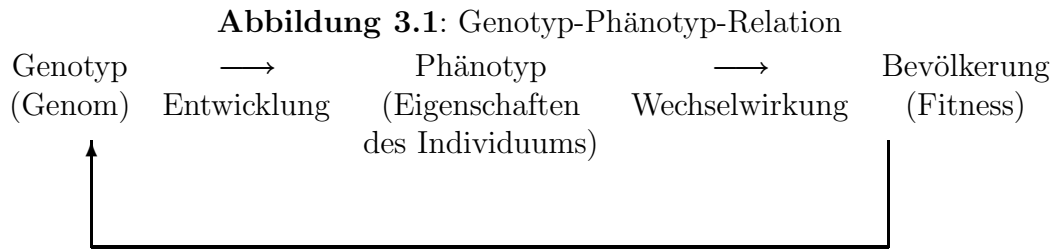
3.1 Die Motivation des Modells

Es existieren zwei wesentliche Motivationen, die bei der Schaffung des Bit-String-Modells Pate gestanden haben. Zum einen beobachtet man in der Biologie in allen Bereichen Hierarchien (Die Bedeutung von Hierarchien für die Bestimmung von Komplexität wurde von HUBERMAN *et al.* [21, 63] hervorgehoben.): Einzelne Zellen haben sich zu mehrzelligen Organismen vereinigt. Mehrere Organe, die selbst schon eine höhere Organisationsstufe darstellen, bilden einen Körper. Dieses Phänomen findet sich aber auch bis in die kleinsten Funktionseinheiten, z.B. die Erbinformation, die aus den DNS-Strängen besteht. Sie ist in der untersten Stufe eine Aneinanderreihung von den vier verschiedenen Nukleotiden Adenin, Guanin, Cytosin und Thymin. Andererseits kann man sie auch als eine Menge von *Genen* betrachten, die jeweils in Teilketten der DNS kodiert sind. Damit sollte das zu erstellende Modell mehrere, aber zumindest zwei Hierarchiestufen besitzen.

Die zweite Motivation besteht darin, einige wesentliche Nachteile bisheriger Modelle biologischer Evolution möglichst zu vermeiden. Die Hauptnachteile anderer präbiotischer Evolutionsmodelle sind die Folgenden:

1. Evolution wird als ein Optimierungsproblem aufgefaßt, das eine wohldefinierte und meistens skalare Zielfunktion besitzt, die sehr viele Minima und Maxima hat. Dies entspricht somit der Annäherung eines Spinglases an seinen Grundzustand, wie schon der Nobelpreisträger ANDERSON [3] erkannte (zur Grundzustandssuche im Spinglas siehe z.B. [39, 40]). Hierzu existieren mehrere Kritikpunkte:
 - (a) Es ist keine biologisch sinnvolle Zielfunktion bekannt.
 - (b) Die Fitness eines einzelnen Individuums hängt nur von der umgebenden Population ab. Sie ist also nur eine lokale Größe und wird von MAYNARD SMITH [91] als *frequency dependent fitness* bezeichnet.
 - (c) Wie bei der Annäherung an ein Spinglas erreicht die Population sehr schnell eines der lokalen Fitnessmaxima und verbleibt dort für längere Zeit, so daß ein metastabiler Zustand, eine dominante Spezies, vorliegt. Je höher ein Fitnessmaximum ist, desto stabiler ist die dazugehörige Spezies. Der Weg von einem Maximum zum nächsten geht ebenfalls über einzelne dominante Spezies, so daß man von Evolution im eigentlichen Sinne nicht reden kann. (Der Begriff Spezies, wie er im Zusammenhang dieser Arbeit verwendet wird, wird im Abschnitt 3.5 noch näher erläutert.)
 - (d) Eine Folge des vorigen Punktes besteht darin, daß die beobachtete Vielfalt des Lebens so nicht erklärt werden kann.
2. Die Anzahl der möglichen Spezies wird als groß aber dennoch beschränkt vorausgesetzt. Damit wird einem wesentlichen Aspekt des Lebens, der Fähigkeit zur Erzeugung immer neuer und unvorhersehbarer Formen, keine Rechnung getragen.

3. Es fehlt die Unterscheidung zwischen Genotyp und Phänotyp, die erst das Wirken der sehr komplexen Rückkopplungsmechanismen erlaubt. Die Genotyp-Phänotyp-Relation wird schematisch in Abbildung 3.1 dargestellt.



Der Genotyp eines Lebewesens ist sein Genom, die Menge aller Gene. Aus diesem entwickelt sich in der Ontogenese der Phänotyp, der die Eigenschaften des Individuums enthält. Hieraus erst ergibt sich über die Wechselwirkung mit den anderen Mitgliedern der Bevölkerung eine Fitness, die sich z.B. in der Zahl der Nachkommen ausdrückt, die selbst wieder Nachkommen zeugen. Damit ergibt sich eine Rückkopplungsschleife auf den Genotyp.

4. Es fehlt der Aspekt der *kombinatorischen Explosion*, d.h. es wird nicht benutzt, daß die Kombination vieler, aber gleichzeitig relativ einfacher Bausteine der wirksamste Weg zur Entstehung eines sehr großen Zustandsraums ist. In diesem Zustandsraum können sich nun komplexe Strukturen bilden, die hierarchisch aufgebaut sind [21, 63]. An dieser Stelle kommt auch der Begriff *Emergenz* ins Spiel, der in Kapitel 7 diskutiert wird.

In dieser Arbeit wird eine Klasse von Modellen betrachtet, die die obigen vier Nachteile zu vermeiden versuchen, die aber ansonsten so einfach wie möglich sind. Diese Modelle beabsichtigen *nicht*, die biologische Realität zu beschreiben. Es wird vielmehr versucht, *Minimalvoraussetzungen* für evolutionäres Verhalten zu finden. Die Biologie sollte dabei nur in soweit berücksichtigt werden, daß sie *Existenzbeweise* liefert; sie ist aber nicht als eigentliches Untersuchungsobjekt gedacht.

3.2 Die Hierarchie

Im Bit-String-Modell werden Individuen durch Bit-Strings, das sind Bit-Ketten, teilweise variabler Länge dargestellt. Dies entspricht der DNS in der Biologie mit dem Unterschied, daß nur zwei *Nukleotide*, nämlich '0' und '1', statt der üblichen vier existieren.

Eine erste nicht-triviale Hierarchiestufe kommt dadurch ins Spiel, daß die Bit-Strings als Aneinanderreihungen von *Genen* aufgefaßt werden, die als funktionelle Gruppen bei der Wechselwirkung von zwei Individuen dienen. Jedes Individuum kann mehrere Gene enthalten (die maximale Anzahl hängt von der Länge in Bits und vom verwendeten Leseschema ab, siehe hierzu speziell Abschnitt 5.1), so daß damit schon mehrere der oben angesprochenen Punkte berücksichtigt sind. Es existieren 20 verschiedene Gene (näheres findet sich im Abschnitt 3.3), so daß eine kombinatorische Explosion schon dann gegeben ist, wenn jedes Individuum nur Platz für 10 Gene enthält. Dies entspricht zwar immer noch einer beschränkten Spezieszahl, diese Schranke liegt aber so hoch (10^{13}), daß sie getrost vernachlässigt werden kann. Außerdem ist so etwas ähnliches wie eine Genotyp-Phänotyp-Relation vorhanden, denn während die genetischen Operatoren auf der Bitebene wirken, findet die Wechselwirkung und die Selektion auf der Genebene statt (dieses wird im Abschnitt 3.4 näher erläutert).

Jedes einzelne Gen wird durch eine charakteristische Bitsequenz dargestellt.

3.3 Die Dekodierung

Eine mögliche Kodierung von Genen wird in Tabelle 3.1 und Abbildung 3.2 dargestellt. Man hat viele Möglichkeiten die prinzipiell gleichwertig sind. Bei der zufälligen Erzeugung des Genbaumes aus Abbildung 3.2 wurde mit einem vollständigen binären Baum mit drei Ebenen gestartet. Dieser Baum enthält dann in der dritten Ebene 8 Knoten. Von da an wurde bei jedem dieser Knoten mit der Wahrscheinlichkeit $P_V = 1/2$ entschieden, ob aus dem Knoten ein Blatt wurde (siehe Abbildung 3.2) oder ob ein Knoten entsteht, der in die nächste Ebene verzweigt. Diese Wahrscheinlichkeit ist die kritische Wachstumswahrscheinlichkeit für einen solchen wachsenden binären Baum. Das in Tabelle 3.1 und in Abbildung 3.2 gezeigte Beispiel ist eine mögliche Realisierung, bei der Gene der Länge 3 bis 8 Bits entstanden sind. Diese Realisierung zeichnet sich dadurch aus, daß sie in den später beschriebenen Simulationen verwendet wurde.

Tabelle 3.1: Die Bitdarstellung des Genbaumes

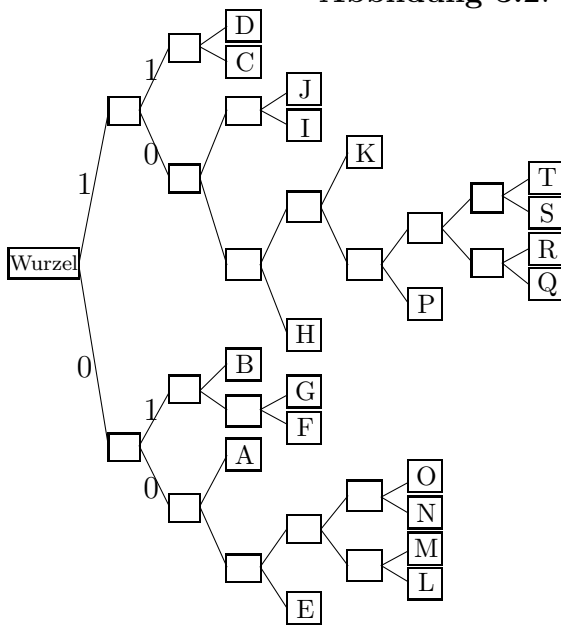
Gen	Länge	Bits	Gen	Länge	Bits
A	3	001	K	5	10011
B	3	011	L	6	000100
C	3	110	M	6	000101
D	3	111	N	6	000110
E	4	0000	O	6	000111
F	4	0100	P	6	100100
G	4	0101	Q	8	10010100
H	4	1000	R	8	10010101
I	4	1010	S	8	10010110
J	4	1011	T	8	10010111
sc	3	110			

Diese Tabelle zeigt die in Abbildung 3.2 als Baum dargestellten Gene als Bit-Ketten. Gene gleicher Länge sind jeweils zu Gruppen zusammengefaßt. Das in Abschnitt 5.5 definierte Startcodon wird durch `sc` gekennzeichnet. (Entnommen aus [42])

Die Dekodierung einer Bitsequenz läßt sich mit Tabelle 3.1 und Abbildung 3.2 leicht erklären. Wenn bei der Dekodierung eines Bit-Strings an einer bestimmten Stelle ein Gen anfangen soll, so befindet man sich an der Wurzel des Genbaumes aus Abbildung 3.2. Jedes Bit wird nun dazu verwendet, sich ebenenweise durch den Baum zu arbeiten. Eine ‘1’ führt in den oberen Zweig, eine ‘0’ in den unteren. Liegt also z.B. die Teilsequenz ‘001’ vor, so startet man an der Wurzel, geht zweimal in den unteren Zweig, danach einmal in den oberen und findet dort ein Blatt, das mit A gekennzeichnet ist. Auf diese Weise ist das Gen A dekodiert worden (vergleiche auch Tabelle 3.1). Wenn man Abbildung 3.2 genau ansieht, stellt man fest, daß jede mögliche Bitsequenz dekodiert werden kann, denn der Baum enthält keine offenen Enden. Dies sieht man nicht so leicht, wenn man nur Tabelle 3.1 zur Verfügung hat. Diese Ausführungen machen auch klar, daß sich somit eine Baumstruktur im Simulationsprogramm als natürliche Struktur bei der Dekodierungsroutine anbietet.

Diese Bit-Strings sind einem *genetischen Algorithmus* mit Selektion, Mutation und Cut-and-Splice unterworfen. Genetische Algorithmen wurden von HOLLAND [61] und unabhängig davon Evolutionsstrategien von RECHENBERG [104] entwickelt, die als Optimierungsverfahren dienen sollen. Beide benutzen Ideen der biologischen Anpassung. Ihre Anwendung im Bit-String-Modell wird im folgenden Abschnitt genauer erläutert.

Abbildung 3.2: Der Genbaum zu Tabelle 3.1



Diese Abbildung zeigt die in Tabelle 3.1 definierten Gene als binären Baum. Offene Quadrate werden als *Knoten* bezeichnet, mit einem Buchstaben gefüllte Quadrate als *Blatt*. Wenn bei der Dekodierung ab einer bestimmten Position des Bit-Strings ein Gen beginnt, so entspricht dies dem Start an der Wurzel (links im Bild). Wird eine ‘1’ vorgefunden, so wird nach oben, bei einer ‘0’ dagegen nach unten in die nächste Ebene verzweigt. Dieses wird solange wiederholt, bis man an ein Blatt gelangt. Da der Baum keine *offenen* Blätter enthält, kann jede mögliche Bitkombination dekodiert werden. (Entnommen aus [42])

3.4 Der verwendete genetische Algorithmus

In dieser Arbeit wird ein einfacher genetischer Algorithmus verwendet. Die Populationsgröße liegt während der gesamten Simulation fest, sie beträgt zwischen 64 und 10 000 Individuen. Zu Beginn wird eine zufällige Anfangspopulation erzeugt, dann finden in jedem Zeitschritt folgende zwei Phasen statt:

1. In der Wechselwirkungsphase erhalten die Individuen eine Bewertung, dieses wird in Abschnitt 3.4.1 näher erläutert. Mit **Wechselwirkung** wird in den folgenden Abschnitten *nur* die Ermittlung der Bewertung (Fitness) bezeichnet.
2. In der Selektionsphase wird entschieden, welche Individuen eliminiert werden und welche Nachwuchs erzeugen. Dies wird in Abschnitt 3.4.2 genauer ausgeführt. Dabei findet auch eine *Wechselwirkung* zwischen Individuen statt. Da diese aber *nicht* die Bewertung verändert, sondern nur auswertet, wird dieses in den folgenden Abschnitten mit **Selektion** bezeichnet. Der Nachwuchs kann, bevor er in die Bevölkerung aufgenommen wird, den genetischen Operatoren Mutation und Cut-and-Splice unterworfen werden. Dieses wird in Abschnitt 3.4.3 behandelt.

Da mit einer konstanten Bevölkerungsgröße gearbeitet wird, kann die Population niemals aussterben. Theoretisch kann mit dem oben skizzierten Schema beliebig lange simuliert werden.

3.4.1 Die Wechselwirkungsphase

In dieser Arbeit wird davon ausgegangen, daß sich die Individuen in einer gut durchmischten Flüssigkeit befinden. Dies hat zur Folge, daß potentiell jedes Individuum mit jedem anderen wechselwirken kann. Zu Beginn des Zeitschritts werden die Bewertungen s_i aller Individuen mit dem Wert 0 initialisiert. Danach wird eine bestimmte Anzahl W von *elementaren Wechselwirkungen* durchgeführt. W ist über die Gleichung

$$W = \frac{1}{2}\alpha N \quad (3.1)$$

mit der Bevölkerungsgröße N verbunden. Die Größe α bezeichnet die mittlere Anzahl der elementaren Wechselwirkungen, an denen jeder Bit-String beteiligt ist. In den in Kapitel 5 vorgestellten Simulationen werden hauptsächlich $\alpha = 10$ und $\alpha = 20$ verwendet.

Jede elementare Wechselwirkung ist eine Zweipunktwechselwirkung, denn es nehmen genau zwei Individuen (i und k) an der Wechselwirkung teil. Bei der Wechselwirkung wird in jedem Individuum ein Gen (g_i und g_k) zufällig ausgewählt und dann die Bewertung der Individuen entsprechend

$$\begin{aligned} s_i &\leftarrow s_i + A(g_k, g_i) \\ s_k &\leftarrow s_k + A(g_i, g_k) \end{aligned} \quad (3.2)$$

geändert. $A(m, n)$ ist eine quadratische Matrix mit zufälligen Einträgen im Intervall $[-1, +1]$. Warum in dieser Arbeit eine zufällige Matrix verwendet wurde, wird in Kapitel 4 noch ausführlich diskutiert. An dieser Stelle soll nur kurz resümiert werden, daß die Komplexität realer biologischer Prozesse es sinnvoll erscheinen läßt, eine zufällige Wechselwirkungsmatrix zu verwenden, solange alle möglichen Wechselwirkungsarten in dieser Matrix repräsentiert sind.

In Tabelle 3.2 wird eine der in Kapitel 5 verwendeten Matrizen dargestellt. Betrachten wir diese nun etwas genauer: Wenn beide Werte, $A(m, n)$ und $A(n, m)$, positiv sind, kooperieren die Gene m und n (siehe z.B. Gene A und F). Wenn beide Werte negativ sind, unterdrücken sich die Gene gegenseitig (siehe z.B. Gene B und F). Wenn ein Wert positiv, der andere negativ ist, wird das eine Gen vom anderen ausgebeutet, d.h., es findet eine Räuber-Beute-Wechselwirkung statt. (Siehe z.B. Gene A und K: Das Individuum, das Gen K enthält, wird von dem, das Gen A enthält, unterdrückt.) Eine neutrale Wechselwirkung ist ebenfalls möglich, wenn beide Matrixeinträge dicht bei Null liegen (siehe z.B. Gene D und J). Bei genauer Betrachtung von Tabelle 3.2 kann man also erkennen, daß jede mögliche Form von Wechselwirkung in dieser Matrix repräsentiert ist.

Tabelle 3.2: Eine verwendete Wechselwirkungsmatrix $A(g_i, g_k)$

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	0.467	0.809	-0.055	-0.039	0.291	0.756	0.258	0.383	-0.768	-0.416	-0.894	0.887	0.710	-0.590	0.638	0.593	-0.546	0.323	-0.166	-0.605
B	0.480	0.849	-0.575	-0.445	-0.112	-0.978	0.874	-0.131	0.828	0.846	0.148	0.961	-0.797	-0.919	-0.904	-0.027	-0.152	-0.089	0.194	-0.203
C	0.270	-0.152	0.647	0.234	-0.543	0.495	-0.485	0.936	-0.067	-0.748	0.410	0.973	0.931	0.970	-0.986	0.034	0.623	-0.759	-0.069	0.860
D	0.110	-0.721	-0.519	0.813	-0.436	0.705	-0.063	0.531	0.931	0.004	-0.725	-0.370	0.946	-0.510	-0.266	-0.531	-0.578	-0.986	-0.864	0.136
E	-0.194	-0.991	0.730	0.452	-0.757	-0.264	-0.461	0.244	-0.355	-0.456	-0.710	-0.241	0.072	0.264	0.313	0.514	0.285	-0.711	0.308	-0.584
F	0.797	-0.917	-0.224	-0.422	0.841	-0.280	0.533	0.381	0.171	0.235	0.470	-0.926	-0.895	0.685	-0.593	0.445	0.662	-0.528	0.813	0.089
G	0.541	-0.854	0.779	0.321	0.004	0.678	0.549	0.850	0.362	0.775	0.150	0.570	0.079	-0.661	-0.865	0.991	-0.691	0.409	0.967	-0.987
H	-0.433	-0.166	-0.606	0.032	-0.817	0.854	-0.292	0.135	0.116	0.062	-0.452	-0.938	0.078	-0.266	0.436	-0.931	-0.583	-0.855	-0.906	-0.293
I	0.463	-0.966	0.234	-0.178	0.432	0.734	-0.007	-0.469	-0.817	0.635	0.615	0.582	0.831	0.052	-0.071	0.415	0.775	0.319	0.762	0.598
J	0.628	0.251	-0.773	-0.039	0.994	-0.337	0.116	0.416	-0.825	0.721	0.637	-0.551	0.880	-0.473	0.700	-0.552	-0.162	-0.013	0.613	-0.178
K	0.834	-0.074	-0.033	-0.137	0.349	-0.294	0.084	0.655	0.803	-0.704	-0.447	-0.690	0.176	0.239	0.225	-0.072	-0.793	-0.947	0.694	-0.728
L	0.545	0.609	-0.194	-0.103	-0.007	-0.119	0.199	0.479	0.769	-0.773	-0.900	-0.732	-0.577	-0.272	-0.633	-0.891	0.608	0.775	-0.619	0.457
M	0.023	0.337	-0.201	-0.763	0.232	-0.977	-0.584	-0.183	0.090	0.977	0.602	0.924	0.008	-0.491	-0.647	0.834	-0.008	0.819	0.301	-0.999
N	0.251	-0.019	0.418	-0.127	-0.328	0.827	-0.724	0.755	0.568	-0.611	0.011	0.032	0.060	0.170	0.470	-0.147	0.926	-0.202	-0.756	-0.379
O	0.468	-0.277	-0.199	-0.290	-0.654	-0.799	-0.191	0.906	-0.143	-0.087	0.685	-0.977	-0.632	0.871	0.906	-0.625	-0.958	-0.364	-0.619	-0.909
P	0.397	-0.414	0.753	-0.196	0.940	0.002	0.486	-0.410	-0.363	0.943	0.764	0.082	0.777	0.366	0.873	0.881	0.177	0.756	0.859	-0.789
Q	0.327	0.324	0.218	0.035	0.343	-0.724	-0.681	-0.993	-0.982	0.335	0.285	-0.222	-0.044	-0.448	0.511	-0.841	-0.734	-0.389	0.754	0.152
R	0.065	-0.861	0.529	0.287	0.650	0.313	-0.478	-0.479	-0.669	0.832	0.614	-0.827	0.753	0.180	-0.234	-0.273	0.042	-0.266	-0.996	0.227
S	-0.711	-0.625	0.106	0.936	-0.612	0.863	0.535	-0.856	-0.361	-0.635	-0.704	-0.286	0.168	0.332	0.036	0.738	0.719	-0.679	0.214	-0.299
T	-0.052	-0.187	-0.300	-0.193	0.858	0.791	-0.981	-0.244	0.168	0.822	0.472	-0.309	-0.693	0.718	0.206	0.223	0.578	-0.959	-0.510	-0.896

Bei der Auswahl der Individuen, die miteinander wechselwirken, wurden zwei verschiedene Methoden verwendet:

1. W Mal wurden zwei Lebewesen völlig zufällig aus der Bevölkerung genommen. Dies führt dazu, daß jeder Bit-String im Mittel an α Wechselwirkungen teilnimmt. Hierbei ergibt sich eine Binomialverteilung um diesen Mittelwert.

2. Man versieht jeden Bit-String mit einer Nummer und erzeugt dann genau α Permutationen dieser Nummern. Man verwendet nun diese Permutationen, um die Wechselwirkungspartner zu erhalten. Auf diese Weise nimmt jeder Bit-String bei einer geradzahigen Bevölkerungsgröße an exakt α elementaren Wechselwirkungen teil.

Der in diesem Abschnitt beschriebene Wechselwirkungsmechanismus ist sehr einfach. Damit genügt er den in Abschnitt 3.1 aufgeführten Kriterien, die an das Bit-String-Modell gestellt werden. Dieser Mechanismus hängt sehr stark von Zufallselementen ab, daß dies sinnvoll und erwünscht ist, wird in Kapitel 4 näher erläutert. Im nächsten Abschnitt werden die verwendeten Selektionsmechanismen vorgestellt.

3.4.2 Die Selektion

Das klassische Vorbild für alle Selektionsmechanismen ist die DARWINSche Selektion. Jedes Individuum erhält entsprechend seiner Fitness s_i Nachkommen, häufig wird dazu eine Fermiverteilung der Form

$$p(s_i) = \frac{\text{constant}}{1 + \exp [(\bar{s} - s_i)/C]} \quad (3.3)$$

verwendet. Dabei ist $\bar{s} = (1/N) \cdot \sum_{i=1}^N s_i$ die mittlere Bewertung und C ein Parameter, der der Temperatur in der statistischen Mechanik ähnelt. Hohe Fitness bedeutet viele Nachkommen, niedrige Fitness wenige oder keine Nachkommen. Verwendet man $C = 0$, so erhalten unterdurchschnittliche Individuen keinen Nachwuchs und alle überdurchschnittlichen Individuen können mit gleicher Wahrscheinlichkeit replizieren. Dieses legt einen ersten sehr einfachen Selektionsmechanismus nahe. Es werden alle unterdurchschnittlichen Individuen entfernt und durch Nachkommen der überdurchschnittlichen Individuen ersetzt. Dieser Mechanismus läßt sich sehr einfach implementieren und er läßt die Bevölkerungsgröße konstant.

Dieser Mechanismus hat allerdings einen bedeutenden Nachteil. Zu Beginn einer Simulation liegen *gute* und *schlechte* Individuen mit etwa gleicher Wahrscheinlichkeit vor. Durch die Selektion wird die Fitness-Verteilung sehr schief, d.h., *schlechte* Individuen tauchen mit wesentlich geringerer Wahrscheinlichkeit auf als *gute*. Die Evolution sorgt dann dafür, daß nur noch eine Spezies vorhanden ist, bei der alle Individuen etwa gleich bewertet sind. Damit existieren kaum noch unterdurchschnittlichen Individuen und der Evolutionsprozeß verlangsamt sich und findet ein schnelles Ende. Deshalb ist es sinnvoller, einen (fast) konstanten Anteil der Bevölkerung zu eliminieren und durch neue Individuen zu ersetzen. Dies führt zu folgende drei Möglichkeiten, die jedes Individuum in einem Zeitschritt hat:

1. Es wird eliminiert.
2. Es wird ohne Veränderung in den nächsten Zeitschritt übernommen, d.h., es überlebt.
3. Es überlebt und zeugt einen Nachkommen, der den genetischen Operatoren unterworfen wird (siehe Abschnitt 3.4.3).

Um dies zu implementieren, gibt es mindestens zwei grundsätzliche Möglichkeiten, die im Folgenden beschrieben werden.

Das Sortiermodell: Die Bit-Strings werden nach ihrer Fitness sortiert. Das schlechteste Viertel wird eliminiert und durch Nachkommen von Individuen aus dem besten Viertel ersetzt. Damit existiert eine konstante Eliminationswahrscheinlichkeit $p = 0.25$. Jedes Individuum bleibt daher im Mittel $\tau = 4$ Zeitschritte in der Simulation. Dies sind die in

dieser Arbeit vorwiegend verwendeten Werte, man kann prinzipiell auch jede andere Eliminationswahrscheinlichkeit p verwenden. In einem Zufallsmodell (siehe Kapitel 4) kann man nun leicht die Individuenaltersverteilung (siehe Anhang A) und die Speziesaltersverteilung berechnen (zur Definition der Spezies siehe Abschnitt 3.5 und zur Berechnung der Speziesaltersverteilung siehe Anhang B).

Das Sortiermodell entspricht einer sehr stark wirkenden Selektion. Nur die in der gesamten Bevölkerung sehr guten Individuen können Nachwuchs zeugen, und die sehr schlechten werden vollständig eliminiert. Dieser *rigide* Selektionsmechanismus hat deutlichen Einfluß auf die Individuenaltersverteilung. Während in einem Zufallsmodell eine Exponentialverteilung beobachtet wird, findet man beim Sortiermodell deutliche Abweichungen (siehe hierzu Kapitel 5 und den Anhang von [42]).

Das Sortiermodell vergleicht die Bewertung eines Individuums *global* mit allen anderen Individuen. Die Wechselwirkung, d.h., die Berechnung der Fitness, findet aber nur mit einem geringen Anteil der Gesamtbevölkerung statt, gewissermaßen *lokal*. Da das Bit-String-Modell ein Modell ohne globalen Controller sein soll (siehe Kapitel 2), sollte dies auch für die Selektion gelten. Das Turniermodell ist ein solches nicht globales Selektionschema.

Das Turniermodell: Beim Turniermodell (englisch *Tournament Selection*) finden Turniere zwischen den Individuen statt. Die Gesamtpopulation wird in zufällige und gleich große Gruppen eingeteilt. Der Beste jeder Gruppe wird in den nächsten Zeitschritt übernommen. Beträgt die Gruppengröße m , so muß jedes Individuum an m Turnieren teilnehmen, damit die Bevölkerungsgröße konstant bleiben kann. Auf diese Weise gewinnt das beste Individuum jedes Turnier, das schlechteste verliert jedes Turnier und ein durchschnittliches gewinnt die Hälfte der Turniere und verliert die Hälfte der Turniere. Dies entspricht grob den Verhältnissen im Sortiermodell. Im Unterschied dazu kann beim Turniermodell ein unterdurchschnittliches Individuum, das gegen noch schlechtere antritt, Nachkommen erzeugen, während ein überdurchschnittliches, das nur gegen bessere antritt, eliminiert werden kann. Damit ist dieser Selektionsmechanismus sanfter.

In dieser Arbeit wird die Turniergröße $m = 2$ verwendet. Ein Individuum,

- das kein Turnier gewinnt, wird eliminiert,
- das ein Turnier gewinnt, wird unverändert in den nächsten Zeitschritt übernommen, d.h, es überlebt,
- das beide Turniere gewinnt, überlebt und zeugt einen Nachkommen, der den genetischen Operatoren unterworfen wird.

Damit läßt sich nun leicht die Eliminationswahrscheinlichkeit p ausrechnen: Man kann jedes Individuum durch eine Bewertung $x \in [0, 1]$ kennzeichnen. Diese Zahl x entspricht der Wahrscheinlichkeit, daß ein beliebiges anderes Individuum eine schlechtere Bewertung hat. Das beste Individuum hat den Wert $x = 1$, das schlechteste den Wert $x = 0$. Damit ergibt sich die Wahrscheinlichkeit, daß ein Individuum mit der Bewertung x eliminiert wird zu

$$q(x) = (1 - x)^2. \quad (3.4)$$

Der Mittelwert von $q(x)$ über die gesamte Bevölkerung ergibt die Eliminationswahrscheinlichkeit p :

$$p = \int_0^1 q(x) dx = \frac{1}{3}. \quad (3.5)$$

Der gleiche Wert ergibt sich für die Wahrscheinlichkeit, daß ein Individuum einen Nachkommen zeugt. Verwendet man eine andere Turniergröße, so läßt sich genauso leicht die

Eliminationswahrscheinlichkeit p berechnen.

Die hier berechnete Eliminationswahrscheinlichkeit p ist natürlich nur ein Mittelwert, denn es kann passieren, daß zweimal dieselbe Einteilung in Gruppen vorliegt, dann gilt in diesem Zeitschritt $p = 0.5$, es kann aber auch passieren, daß nur das schlechteste Individuum eliminiert wird, da jedes andere ein Turnier gewinnt und eins verliert, dann ist $p = 1/N \simeq 0.0$.

Ein dritter Selektionsmechanismus, das Energieflußmodell, wurde von WOLTER in seiner Diplomarbeit [119] behandelt und wird in Abschnitt 3.6 kurz vorgestellt.

3.4.3 Die genetischen Operatoren

Mutationen: Mutationen werden dazu verwendet, neues genetisches Material in die Bevölkerung einzuführen. Man kann dies auf verschiedene Weisen realisieren [47]. In dieser Arbeit wurden zwei Methoden verwendet.

Bei einem Individuum, das Nachwuchs erzeugt, findet **eine** Punktmutation statt, dies wird auch als *single-bit flip* bezeichnet, d.h., **ein** Bit wird von 0 auf 1 bzw. von 1 auf 0 umgeklappt. Dies erzeugt bei einer Länge der Bit-Strings von 96 Bits eine Mutationsrate von etwa 1% pro Bit. Die Anwendung dieses Operators ist schnell und einfach, denn pro replizierendem Bit-String wird nur eine Zufallszahl benötigt. Ein Nachteil dieser Implementation besteht darin, daß bei einigen der in Kapitel 5 vorgestellten Dekodiermechanismen ein Individuum keine identischen Nachkommen zeugen kann. Außerdem ist die Mutationsrate von einem Prozent pro Bit ziemlich hoch. Bei kürzeren Bit-Strings und ebenfalls single-bit flip entstehen sogar noch höhere Mutationsraten.

Will man eine niedrigere Mutationsrate p_{mut} (z.B. 10^{-3}) verwenden, muß man andere Wege gehen. Eine zugegebenermaßen sehr zeitaufwendige Methode besteht darin, bei der Replikation bei jedem Bit zu entscheiden, ob ein Bitflip stattfindet oder nicht. Dieses hat den Nachteil, daß man sehr viele Zufallszahlen ziehen muß, obwohl dann schließlich nur sehr wenige Bits umgeklappt werden. Eine schnellere Methode, die auf der gerade beschriebenen basiert, verwendet die Wahrscheinlichkeiten $P(n)$, daß bei vorgegebenem p_{mut} genau n Punktmutationen stattfinden. Es gilt:

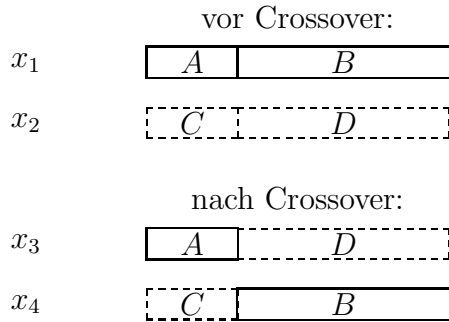
$$P(n) = \binom{l}{n} (1 - p_{mut})^{l-n} p_{mut}^n, \quad (3.6)$$

wobei l die Länge des Bit-Strings in Bits ist. Berücksichtigt man nur die Wahrscheinlichkeiten, für die z.B. $P(n) > 10^{-8}$ gilt, so können bei $l = 96$ und $p_{mut} = 10^{-3}$ (10^{-2}) maximal 5 (10) Punktmutationen stattfinden. Die vernachlässigten Werte haben zusammen eine Wahrscheinlichkeit $P(n > 5) < 6 \cdot 10^{-8}$ ($P(n > 10) < 3 \cdot 10^{-8}$). Damit kann man im Mittel mit 1.1 (2.0) Zufallszahlen pro Replikationsvorgang zwischen 0 und 5 (10) Punktmutationen realisieren.

Beide Varianten des Mutationsoperator wurden bei den Simulationen, die in Kapitel 5 diskutiert werden, verwendet.

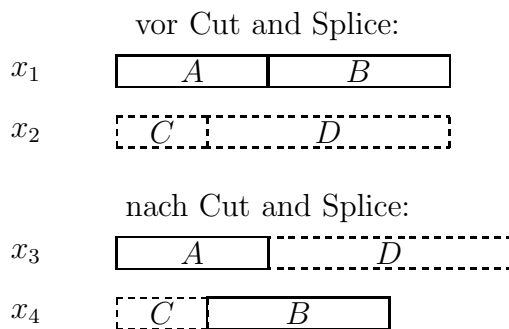
Cut-and-Splice: Ein Operator, der bei sexueller Reproduktion verwendet wird, ist Crossover. Eine übliche Implementation sieht vor, daß zwei Individuen (x_1 und x_2) an derselben Stelle zerschnitten werden; dann werden zwei neue Individuen (x_3 und x_4) zusammengesetzt, die jedes eine Mischung aus den Ursprungsindividuen sind (siehe Abbildung 3.3). Crossover wird vor allem bei Optimierungsproblemen angewendet, wenn das zu optimierende Problem auf einen Bit-String einer genau festgelegten Länge abgebildet worden

Abbildung 3.3: Crossover



Diese Abbildung zeigt schematisch, wie aus zwei Individuen gleicher Länge durch die Anwendung von Crossover zwei neue Individuen ebenfalls gleicher Länge entstehen. Siehe hierzu z.B. [61]. (Entnommen aus [119])

Abbildung 3.4: Cut-and-Splice



Diese Abbildung zeigt schematisch, wie aus zwei Individuen, in diesem Fall gleicher Länge, durch die Anwendung von Cut-and-Splice zwei neue Individuen unterschiedlicher Länge entstehen. Ein Spezialfall von Cut-and-Splice ist der Crossover-Operator, der die Länge der Individuen erhält (siehe Abbildung 3.3). (Entnommen aus [119])

ist. In dieser Arbeit existiert aber nicht die Notwendigkeit einer konstanten Länge der Bit-Strings, prinzipiell kann ein Bit-String eine beliebige Länge haben. Damit ist es nicht notwendig einen genetischen Operator zu verwenden, der die Länge erhält. Als ein solcher Operator wurde von GOLDBERG *et al.* [48, 49] Cut-and-Splice eingeführt. Abbildung 3.4 zeigt seine Wirkungsweise. Man erkennt sofort, daß Crossover nur ein Spezialfall von Cut-and-Splice ist. Einige Simulationen haben Cut-and-Splice verwendet, in Kapitel 5 wird jeweils besonders darauf hingewiesen. Die meisten Simulationen haben allerdings nur einen der oben beschriebenen Mutationsoperatoren verwendet.

3.5 Die Meßgrößen

Nach den bisher gemachten Ausführungen ist man in der Lage, ein Simulationsprogramm zu schreiben, das Daten erzeugt. Nun ist es wichtig, welche Meßgrößen man als geeignet auswählt. Dieses stellt einen nicht trivialen Aspekt dieser Arbeit dar. Es wird versucht, einen Prozeß zu starten, der der biologischen Evolution ähnelt. Dort hat eine Selbstorganisation von einfachen Bausteinen zu komplexen Strukturen stattgefunden. Im Bit-String-Modell sind einfache Bausteine, nämlich die Bit-Strings, die aus Genen bestehen, vorhanden. Wie könnten nun komplexe Strukturen aussehen? Dies ist sicher a priori nicht vorhersagbar. Daher ist es wichtig, Meßroutinen gut zu planen und neuen Erkenntnissen immer wieder anzupassen. Im Folgenden werden einige der verwendeten Meßgrößen vorgestellt.

Die Altersverteilung: Jedes Individuum kann mehrere Zeitschritte in der Simulation verbleiben. In dem in dieser Arbeit verwendeten Sortiermodell beträgt die mittlere Lebenserwartung $\tau = 4$ Zeitschritte, im Turniermodell nur 3 Zeitschritte. Mittelt man über eine komplette Simulation von typischerweise $t_{sim} = 10\,000$ Zeitschritten, so erhält man

eine Verteilung. Ist man in der Lage, diese Verteilung zu verstehen und vielleicht sogar herzuleiten? Hier erweisen sich Zufallsmodelle (siehe auch Kapitel 4) als sehr wichtiges Handwerkszeug. In Anhang A wird gezeigt, daß sich für die Altersverteilung der Individuen eine Exponentialverteilung ergibt. Abweichungen von dieser Verteilung können auf den Einfluß der Selektion zurückgeführt werden, da *fittere* Individuen länger in der Simulation verbleiben und *unfittere* Mutanten sehr schnell eliminiert werden. (Andererseits kann eine Abweichung auch auf einen logischen Fehler hinweisen.)

Die Altersverteilung ist eine globale Meßgröße, denn sie kann bei einer endlichen und sogar sehr kleinen Bevölkerungsgröße nur dann mit hoher Statistik ermittelt werden, wenn man über sehr lange Zeiträume mißt. Evolution ist aber ein dynamischer Prozeß, daher sollten auch (zeitlich) lokale Meßgrößen existieren, deren Zeitreihen Aufschluß über die Dynamik geben können.

Mittlere Bewertung: Da die Bit-Strings in der Wechselwirkungsphase eine Bewertung erhalten, die zu Beginn jedes Zeitschritts initialisiert wird, handelt es sich bei der mittleren Bewertung um eine zeitlich lokale Größe. Da die hier vorgestellten Simulationen immer mit einer zufälligen Population beginnen und da alle verwendeten Wechselwirkungsmatrizen Werte aus dem symmetrischen Intervall $[-1, +1]$ enthalten, sollte die mittlere Bewertung bei Null beginnen. Die Selektion sollte dann dafür sorgen, daß die mittlere Bewertung ansteigt. In den in Kapitel 5 vorgestellten Simulationen wird die Bewertung unabhängig von der wirklichen Wechselwirkungsrate α immer auf 10 elementare Wechselwirkungen pro Individuum normiert dargestellt. Damit kann die mittlere Bewertung niemals beliebig hoch ansteigen. Da kein globales Optimierungsziel vorgegeben wird, sondern immer nur intrinsische Adaption an die lokale Population stattfindet, muß die Bewertung auch nicht monoton ansteigen. Sie kann Einbrüche erleben und könnte sogar unter Null absinken.

Die Genbesetzungszahlen: Da die Gene die einzigen aktiven Elemente im Bit-String-Modell sind, sind die Genbesetzungszahlen eine natürliche Meßgröße. Hierbei zeigt sich die Dominanz einiger Gene. Betrachtet man nun noch die Bit-Strings genauer, die relativ alt werden, z.B. älter als 3τ , so erkennt man, daß dort einige Genkombinationen immer wieder auftauchen. Dies führt dann ganz natürlich dazu, daß man den Begriff der Spezies einführt.

Die Spezies: Wie in Abschnitt 3.4.1 erläutert wurde, werden die Gene in einer elementaren Wechselwirkung zufällig ausgewählt. Betrachten wir nun z.B. einen Bit-String, der die Gene $DDEQ$ enthält: Bei einer mittleren Wechselwirkungsrate von $\alpha = 20$ nimmt Gen D an 10 und Gene E und Q jeweils an 5 Wechselwirkungen teil. Dasselbe gilt aber auch für einen Bit-String der Form $QDEQDDED$. Obwohl dieser Bit-String doppelt so viele Gene und dazu noch in einer anderen Reihenfolge enthält, sollten beide Bit-Strings bis auf Fluktuationen dieselbe Bewertung erhalten. (Ein schlechter Zufallszahlengenerator kann diese Annahme ungültig machen, bei dem in dieser Arbeit verwendeten Generator vom KIRKPATRICK-STOLL Typ [74] ist dies nicht zu erwarten.) Wenn die Wechselwirkungsrate beliebig groß wird, sind zwei Individuen, die zur selben Spezies gehören, nicht unterscheidbar. Dies führt zu der folgenden Definition einer Spezies:

- Alle Individuen mit stöchiometrisch gleichen Geninhalten werden zu **einer** Spezies zusammengefaßt. Diese wird durch eine geordnete Kette von Buchstaben gekennzeichnet. Verwendet man eine chemische Schreibweise (die beiden obigen Individuen gehören zur Spezies D_2EQ), so haben die Indizes (nicht vorhandene Indizes haben den Wert 1) als größten gemeinsamen Teiler den Wert 1.

Das Konzept der Spezies eröffnet neue Meßgrößen, diese sind

- die Speziesaltersverteilung,
- die Anzahl der verschiedenen Spezies,
- die Besetzungszahl der dominanten Spezies,
- die Anzahl *alter* Spezies, wobei *alt* noch näher definiert werden muß, und
- die Anzahl *erfolgreicher* Spezies, wobei eine Spezies als erfolgreich gilt, wenn ihre Besetzungszahl einen noch näher zu spezifizierenden Wert überschreitet.

3.6 Das Energieflußmodell

Eine mögliche Variante des Bit-String-Modells, die von WOLTER im Rahmen seiner Diplomarbeit untersucht wurde [119], sieht vor, keine explizite Fitness zu verwenden, sondern einen Wettbewerb um beschränkte Ressourcen (in diesem Fall *Energie*) einzuführen. Dem Gesamtsystem wird pro Zeitschritt eine konstante (quasi-konstante) Menge Energie zugeführt. Die Individuen verbrauchen pro Zeitschritt zusammen etwa dieselbe Menge Energie, wobei jedes Individuum pro Zeitschritt fest vorgegebene Energiekosten hat. Das System wird durch diesen Energiefluß in Gang gehalten.

Individuen, die viel Energie *einsammeln*, können Nachwuchs zeugen, andere, die wenig oder keine Energie einsammeln, sterben. Damit ähnelt dieses Modell dem Evolving-Bugs-Modell von PACKARD [4, 11, 96], das in Kapitel 6 ausführlich beschrieben wird. Während bei PACKARD eine explizite, räumliche Umgebung existiert, arbeitet WOLTER ohne diese. Es existiert ein Energiereservoir, daß zu Beginn jedes Zeitschritts gefüllt wird. In der Wechselwirkungsphase erhalten die Individuen Energie entsprechend den Einträgen der Wechselwirkungsmatrix A , die in diesem Fall nur ganzzahlige Einträge im Intervall $[0, 5000]$ enthält. Wenn eine Wechselwirkung mehr Energie anfordert, als im Reservoir noch vorhanden ist, nimmt das erste, bei WOLTER als *aktives* bezeichnete Individuum dem zweiten (dem *passiven*) Energie weg. Auf diese Weise können kooperative Wechselwirkungen stattfinden, solange noch Energie im Reservoir vorliegt, aber auch Räuber-Beute-Wechselwirkungen, wenn das Reservoir leer ist.

Ein Problem dieses Modells besteht darin, daß neue Variablen eingeführt werden, die starken Einfluß auf die Dynamik haben können. Während das in der vorliegenden Arbeit verwendete Modell eine konstante Bevölkerungsgröße hat und damit ein Aussterben der Bevölkerung nicht möglich ist, schwankt bei dem Energieflußmodell die Bevölkerungsgröße, so daß eine ungünstige Wahl der Parameter dazu führen kann, daß die Population ausstirbt.

Andererseits läßt sich das Energiekonzept leichter biologisch motivieren, als das abstrakte Fitnesskonzept. Da das Fitnesskonzept aber weniger freie Parameter hat, ist es einfacher und leichter analytischen Rechnungen zugänglich, wie sie in Anhang A und B durchgeführt werden.

Die Untersuchungen in [119] zeigen aber, daß die analytischen Rechnungen auch im Energieflußmodell noch verwendet werden können, wenn man Mittelwerte statt der konstanten Größen verwendet. Im übrigen zeigen die Untersuchungen von WOLTER, daß die in dieser Arbeit gefundenen Ergebnisse auch beim Energieflußmodell ihre Gültigkeit behalten. Außerdem hat sich auch in [119] die Arbeit mit Zufallsmodellen (siehe hierzu speziell Kapitel 4) als wichtiges Handwerkszeug erwiesen.

4 Die Bedeutung von Zufall

In allen Artificial Life Simulationen wird sehr stark Gebrauch von Zufallszahlen gemacht. Es treten dabei unterschiedliche Aspekte von Zufall auf. Dies soll speziell am Beispiel des Bit-String-Modells näher erläutert werden. Einige Aspekte der folgenden Diskussion wurden schon in [42] veröffentlicht.

4.1 Unterschiedliche Aspekte des Zufalls

Im Bit-String-Modell, das in Kapitel 3 vorgestellt wurde, tritt Zufall neben der zufälligen Anfangspopulation in drei Formen auf. Diese wurden in Abschnitt 3.4.1 bei der Beschreibung der Wechselwirkungsphase erklärt und werden im Folgenden noch einmal zusammengefaßt:

1. die Wechselwirkungsmatrix $A(g, g')$,
2. die Wahl der Wechselwirkungspartner, und
3. die Wahl der wechselwirkenden Gene g und g' in den Wechselwirkungspartnern.

Die Zufälligkeit der Wechselwirkungsmatrix ist *eingefroren*, denn sie wird zu Beginn einer Simulation erzeugt und danach nicht mehr verändert. Im Gegensatz dazu sind die beiden anderen Zufallsaspekte dynamisch, die Auswahl wird zu jedem Zeitschritt bzw. in jeder elementaren Wechselwirkung neu getroffen. Im Folgenden werden einige Gründe genannt, die für die unterschiedlichen Aspekte des Zufalls sprechen.

zu 1.: Betrachten wir zunächst einmal die Wechselwirkungsmatrix. In der Biologie ist die Wechselwirkung zwischen Genen ein sehr komplizierter Prozess, der die Faltung der DNS selbst, die Faltung der daraus entstandenen Proteine, den Einfluß auf den Phänotyp, die Wechselwirkung zwischen den Phänotypen und noch viele andere Aspekte enthält. Alle diese Schritte sind höchstwahrscheinlich sehr sensitiv auf kleine Details. Es ist weder durchführbar noch wünschenswert, solch eine Dynamik detailliert zu behandeln. Statt explizit zwischen Genotyp und Phänotyp zu unterscheiden und statt eine deterministische Vorschrift für die Wechselwirkung zwischen Genen zu entwerfen (siehe z.B. FONTANA, SCHNABL und SCHUSTER [38]), erscheint es sinnvoll, eine zufällige Wechselwirkungsmatrix zu verwenden, die alle in der Wirklichkeit möglichen Wechselwirkungen enthält. Welche Wechselwirkungstypen möglich sind und wie dies in der Matrix repräsentiert wird, ist bei der Erklärung des Wechselwirkungsmechanismus in Abschnitt 3.4.1 schon ausführlich erläutert worden.

zu 2.: Die Erklärung für den zweiten Aspekt, die zufällige Wahl des Wechselwirkungspartners, ist auch sehr einfach. Sie wurde ebenfalls schon in Abschnitt 3.4.1 genannt: Die Umgebung im Bit-String-Modell wird als eine gut durchmischte Flüssigkeit angesehen, in der die einzelnen Individuen frei herumschweben. Daher wird eine Wechselwirkung zwischen zwei Individuen am besten wiedergegeben, wenn man sie zufällig auswählt.

zu 3.: Betrachten wir nun den dritten Punkt, die Auswahl der wechselwirkenden Gene. Wenn wir uns einen Bit-String als eine eindimensionale Schnur in einer dreidimensionalen Flüssigkeit vorstellen, so ist es schwer nachvollziehbar, daß bei einer Kollision die Bit-Strings immer parallel aufgereiht aufeinandertreffen. Es ist wahrscheinlicher, daß nur jeweils kleine Teile miteinander in Kontakt kommen. In dieser Arbeit werden diese Teile als die Gene betrachtet. Wenn man die Bit-Strings als dreidimensionale Strukturen betrachtet, so ist die Argumentation ähnlich. Nur Teile auf der Oberfläche dieser Struktur

sind aktiv. Auch dies führt dazu, daß bei einer Kollision nur wieder zwei Gene miteinander wechselwirken.

Als letzter Aspekt soll noch einmal der Vergleich zur Wechselwirkung echter biologischer Systeme herangezogen werden. Wenn man davon ausgeht, daß die Wechselwirkung durch eine Art Lesemechanismus durchgeführt wird, dann sind für einen außenstehenden Beobachter die Effekte dieses Mechanismus sehr komplex und nicht leicht nachvollziehbar. Sie erscheinen wie zufällig. Somit ist es naheliegend, mit all diesen Zufallselementen, die in diesem Abschnitt noch einmal zusammengefaßt wurden, zu arbeiten.

Dennoch soll an dieser Stelle auf mögliche Alternativen zu den Zufallsaspekten kurz eingegangen werden.

4.2 Alternative Implementationen

Obwohl im vorigen Abschnitt stichhaltige Gründe für die einzelnen Aspekte des verwendeten Zufalls gegeben wurden, kann man das Bit-String-Modell dafür kritisieren, daß zuviel Zufall vorhanden ist. Deshalb sollen in diesem Abschnitt mögliche Alternativen diskutiert werden.

Will man die Wechselwirkungsmatrix durch irgendetwas Deterministisches ersetzen, so hat man das Problem, daß man eine eigene *Chemie* entwerfen muß, die die Grundbausteine der Wechselwirkungen enthält. So etwas wurde von FONTANA [37] mit seiner *Algorithmic Chemistry* versucht. Dieses Modell ist zwar sehr interessant, aber es übersteigt den Rahmen dieser Arbeit, ein solches Unterfangen in Angriff zu nehmen. Außerdem scheint das Modell in [37] zu *streng* deterministisch zu sein, wie schon in Abschnitt 2.1 kurz erläutert wurde. In der Matrix $A(g, g')$ sind alle möglichen Wechselwirkungen vorhanden; wie diese implementiert sind, spielt keine Rolle, um ihren Einfluß auf die Dynamik eines Systems wechselwirkender Bit-Strings zu untersuchen.

Will man die Auswahl der Wechselwirkungspartner deterministisch machen, muß man sich mögliche Mechanismen überlegen. Eine natürliche Auswahl besteht darin, die Wechselwirkung nächster Nachbarn einzuführen. Dazu muß aber eine Nachbarschaft definiert werden, die wiederum von der Definition einer Metrik abhängt. Hierzu muß bemerkt werden, daß die räumliche Struktur eines Systems sehr starken Einfluß auf die Dynamik haben kann. Dies erkennt man am Beispiel des Evolving-Bugs-Modells von PACKARD [4, 11, 96], das in Kapitel 6 ausführlich vorgestellt wird. Da im Bit-String-Modell die Wechselwirkung im Vordergrund stehen soll, ist es sinnvoll, das Konzept einer gut durchmischten Flüssigkeit zu verwenden. Dies verhindert, daß die Wechselwirkungseffekte von räumlichen Artefakten gestört werden. Ein einfaches System, das nur räumliche Effekte ohne eine explizite Wechselwirkung und ohne explizite Zufallselemente betrachtet, wird in Anhang C ausführlich diskutiert. Ähnlich wie bei chaotischen Systemen wird dort zufälliges Verhalten spontan erzeugt [41].

Betrachten wir nun noch einmal genauer den dritten Aspekt des Zufalls im Bit-String-Modell, die Auswahl der wechselwirkenden Gene. Obwohl Gründe dafür angegeben wurden, eine zufällige Auswahl zu treffen, sollen nun kurz drei andere Möglichkeiten in Augenschein genommen werden. Es seien jeweils die Individuen i und k als Wechselwirkungspartner ausgewählt:

a) Jedes Gen von i wechselwirkt mit jedem Gen von k , die resultierende Änderung der Bewertung wird auf eine Wechselwirkung normiert. Die Normierung verhindert, daß Individuen mit mehr Genen unabhängig von der *Qualität* der Gene im Vorteil gegenüber

solchen mit wenigen Genen sind.

Diese Implementation liefert qualitativ ähnliche Ergebnisse wie die, die in Kapitel 5 vorgestellt werden. Gleichzeitig dauert eine elementare Wechselwirkung wesentlich länger, denn es muß auf wesentlich mehr Einträge der Wechselwirkungsmatrix zugegriffen werden. Bei 10 Genen pro Individuum werden 100 statt nur ein Eintrag der Wechselwirkungsmatrix abgefragt. Dies kann sicher kein Hauptkriterium gegen diese Implementation sein.

Einen anderen Gesichtspunkt liefert die Abschätzung der Wechselwirkungszeit. Wenn man annimmt, daß pro Wechselwirkung in *realen* Systemen eine Zeit T_w nötig ist, so hat ein Gen in einer realistischen Situation gar nicht genug Zeit, mit allen anderen Genen wechselzuwirken. Wenn jedes Gen mit jedem anderen wechselwirkt, dann wird bei langen Individuen mehr Zeit verbraucht als bei kürzeren. Wenn ein Zeitschritt der Simulation einem festen Zeitintervall entsprechen soll, ist diese Implementation sicher nicht die günstigste.

b) Eine Abwandlung des vorigen *vollständigen* Wechselwirkungsmechanismus läßt jedes Gen des Individuums i mit genau einem anderen Gen von k wechselwirken. Hier muß man aber die Frage stellen: Welches Gen von i wechselwirkt mit welchem Gen von k ? Eine natürliche Antwort ist es, die Bit-Strings nebeneinanderzulegen. Das erste Gen von i wechselwirkt mit dem ersten Gen von k , das zweite von i mit dem zweiten von k usw. Dies kann man sich als eine simultan und parallel durchgeführte Aktion vorstellen.

Wie behandelt man aber den sehr häufig auftauchenden Fall unterschiedlicher Längen (in Genen gemessen)? Wird das kürzere bestraft, wird das längere bestraft, d.h., welche Normierung wird verwendet? Dieses Problem läßt sich prinzipiell lösen, dabei müssen aber neue Parameter eingeführt werden, die die bisherige Einfachheit des Modells stören.

c) Eine letzte Möglichkeit, die in diesem Abschnitt behandelt werden soll, versucht, die Information darüber, welche Gene wechselwirken sollen, im Individuum selbst abzuspeichern, d.h., die Liste der Wechselwirkungen wird genetisch kodiert. Eine mögliche Realisierung wurde implementiert, deren Ergebnisse hier kurz skizziert werden.

Die Länge eines Individuums betrug bei dieser Implementation maximal 16 Gene, also werden die ersten vier Bits des Bit-Strings i dazu verwendet, die Nummer des *aktiven* Gens in i zu speichern. Die nächsten vier Bits in i werden dazu verwendet, die Nummer des *passiven* Gens in dem Wechselwirkungspartner k zu speichern. Auf diese Weise entsteht eine unsymmetrische Wechselwirkung, bei der ein aktives und ein passives Individuum existieren. Es kann nun sein, daß keins der beiden Gene vorhanden ist, weil beide Bit-Strings wenige Gene enthalten, oder daß eines der Individuen dieses Gen nicht enthält. Fehlt dem *aktiven* Individuum das *aktive* Gen, so wird es selbst bestraft:

$$s_i \leftarrow s_i - 1.0, \quad (4.1)$$

fehlt analog dem *passiven* Individuum das *passive* Gen, so wird dieses bestraft:

$$s_k \leftarrow s_k - 1.0. \quad (4.2)$$

Diese Bestrafung soll verhindern, daß es günstiger sein kann, eine Wechselwirkung zu vermeiden. Die Wechselwirkung kann wieder dadurch symmetrisch gemacht werden, daß eine elementare Wechselwirkung aus zwei Phasen besteht. In der ersten Phase ist Individuum i aktiv und k passiv, in der zweiten ist es umgekehrt.

Eine Folge dieser Implementation ist, daß zunächst eine Optimierung der ersten acht Bits jedes Bit-Strings eingeleitet wird: Die ersten vier Bits, die das aktive Gen enthalten, müssen einen niedrigen Wert kodieren, damit sich ein kurzes Individuum keinen persönlichen Nachteil bei einer Wechselwirkung einhandelt. Die zweiten vier Bits müssen einen

hohen Wert kodieren, damit das aktive Individuum, wenn es auf ein kurzes Individuum trifft, Vorteile für sich daraus ziehen kann. Dies wiederum zieht dann eine Entwicklung zu Individuen mit vielen Genen nach sich.

Da viele der Gene gar nicht an Wechselwirkungen teilnehmen, weil sie in der Mitte des Bit-Strings stehen, hat das Konzept der Spezies, wie es in Abschnitt 3.5 eingeführt wurde, keinen Sinn. Um doch noch Spezies definieren zu können, kann man Gebrauchszähler einführen, wie sie in Abschnitt 6.3.2 vorgestellt werden. Diese erlauben es, eine *effektive* Spezies zu ermitteln. Die Wechselwirkung zwischen den Genen steht dann erst nach der Optimierung der Kodierungsbits im Vordergrund.

Da in dieser Arbeit die Wechselwirkung selbst das Forschungsobjekt sein soll, ist in den in Kapitel 5 vorgestellten Simulationen keine der Alternativen, die in diesem Abschnitt vorgestellt wurden, berücksichtigt worden.

4.3 Das Zufallsmodell

In den ersten beiden Abschnitten dieses Kapitels steht die im Bit-String-Modell verwendete Zufälligkeit im Mittelpunkt. Trotz dieser Zufälligkeit liegt ein System mit wirksamer Selektion vor. Wie kann man den Einfluß der Selektion genau feststellen? Man betrachtet ein Modell, in dem keine Selektion wirkt, dies wird auch als neutrale Selektion bezeichnet. DERRIDA und PELITI [27] bezeichnen die Fitnesslandschaft in diesem Fall als flach (flat fitness landscape).

In dieser Arbeit wird das Modell mit neutraler Selektion als *Zufallsmodell* bezeichnet. Die gesamten Simulations- und Auswerterroutinen können übernommen werden, nur die Wechselwirkungsroutine muß verändert werden. Eine Möglichkeit besteht darin, jedem Individuum willkürlich eine Zufallszahl als Bewertung zuzuweisen. Danach kann die Selektion wie im Modell mit *normaler* Selektion stattfinden. Da die Bewertung nicht vom Geninhalt abhängt, können einfache statistische Überlegungen zur Individuenaltersverteilung und zur Speziesaltersverteilung durchgeführt werden, wie dies in den Anhängen A und B gezeigt wird.

Der Vergleich von Daten aus Zufallsmodellen mit Selektionsmodellen zeigt, wo die Selektion wirkt und wie stark ihr Einfluß ist. Die Simulationen in Kapitel 5 zeigen zum Beispiel, daß bei dem in Abschnitt 3.4.2 besprochenen Sortiermodell die Selektion auf einzelne Individuen sehr stark wirkt, da die Individuenaltersverteilung sehr deutlich von den Vorhersagen aus dem Zufallsmodell abweicht. Beim ebenfalls in Abschnitt 3.4.2 vorgestellten Turniermodell sind die Abweichungen geringfügiger, da das Zufallselement bei diesem Modell stärker zum Zuge kommt. Die Speziesaltersverteilung weist aber bei beiden Selektionsmechanismen eine Form auf, die kraß von den Vorhersagen des Zufallsmodells abweicht.

Zusammenfassend läßt sich sagen, daß der große Wert von Zufallsmodellen darin besteht, Evolutionsexperimente statistischen Vorhersagen zugänglich zu machen. Dieses hilft zum einen, Evolutionsprozesse besser verstehen zu können (siehe Kapitel 5 und 6). Dabei kann es sogar passieren, daß falsche Konzepte entdeckt werden (siehe hier vor allem Kapitel 6 und speziell Abschnitt 6.3.3). Zum anderen helfen Zufallsmodelle auch, Fehler in Programmen zu finden. Der zweite Aspekt ist bei der Erstellung von komplexen Programmen besonders wichtig, auch wenn es sich dabei *nur* um ein technisches Problem handelt.

5 Simulationen des Bit-String-Modells

Das in Kapitel 3 definierte Bit-String-Modell ist noch nicht eindeutig, denn es beschreibt eine ganze Klasse von Modellen. Die in Abschnitt 3.3 beschriebene Dekodierung eines Bit-Strings wird erst dann eindeutig, wenn geklärt wird, an welchen Stellen Gene beginnen. Es gibt unterschiedliche Möglichkeiten, diese Stellen festzulegen. Dies definiert den jeweiligen Leserahmen.

5.1 Verschiedene Leseschemata

In dieser Arbeit werden drei unterschiedliche Leseschemata verwendet. In diesem Abschnitt werden sie kurz charakterisiert, bevor sie in den weiteren Abschnitten ausführlich diskutiert werden. Einige der dabei vorgestellten Ergebnisse wurden schon in [42] veröffentlicht:

1. Beim ersten Leseschema werden die Bit-Strings als zirkular geschlossen aufgefaßt (zur biologischen Relevanz dieses Ansatzes siehe [28]). An jeder Stelle des Bit-Strings beginnt ein Gen. Dies hat zur Folge, daß ein Bit-String von 32 Bits auch 32 Gene enthält. Es liegt also eine sehr *hohe* Dichte der Gene vor und jedes Gen überlappt mit mehreren anderen Genen. Simulationsergebnisse werden in Abschnitt 5.3 diskutiert.
2. Um das Überlappen zu verhindern, kann man ein neues Gen erst dann anfangen lassen, wenn das vorherige beendet ist. Auf diese Weise wird ein (nicht zirkular geschlossener) Bit-String von vorne nach hinten dekodiert und es liegt das Dekodierschema *Gen an Gen* vor. Hierbei gehört jedes Bit zu genau einem Gen. (Ausnahmen können die letzten 1 bis 7 Bits des Bit-Strings bilden.) Ein Bit-String von 32 Bits enthält etwa 6 bis 10 Gene, was einer *mittleren* Dichte von Genen entspricht. Dieses Modell wird in Abschnitt 5.4 diskutiert.
3. Ein drittes Leseschema verwendet sogenannte *Startcodons*. Diese sind, analog zur Biologie (siehe z.B. [29, 75]), kurze Sequenzen, die kennzeichnen, an welcher Stelle ein Gen beginnt. Ein Bit-String wird von vorne nach hinten auf das verwendete Startcodon hin untersucht und hinter jedem Startcodon fängt ein Gen an. In dieser Arbeit wurde als Startcodon die Sequenz ‘110’ verwendet (siehe auch Tabelle 3.1). Diese Startsequenz hat bei einem zufälligen Bit-String die Wahrscheinlichkeit 2^{-3} , an einer beliebigen vorgegebenen Stelle vorzuliegen. Dies hat zur Folge, daß ein Bit-String von 32 Bits im Durchschnitt 4 Startcodons und damit auch etwa 4 Gene enthält, was einer *niedrigen* Dichte von Genen entspricht. Die Anzahl der Startcodons ist nicht notwendigerweise identisch mit der Anzahl der dekodierten Gene, denn ein Startcodon, das sehr weit hinten im Bit-String vorliegt, muß nicht mehr genug Bits für ein weiteres Gen hinter sich haben. Dieses Modell wird in Abschnitt 5.5 diskutiert.

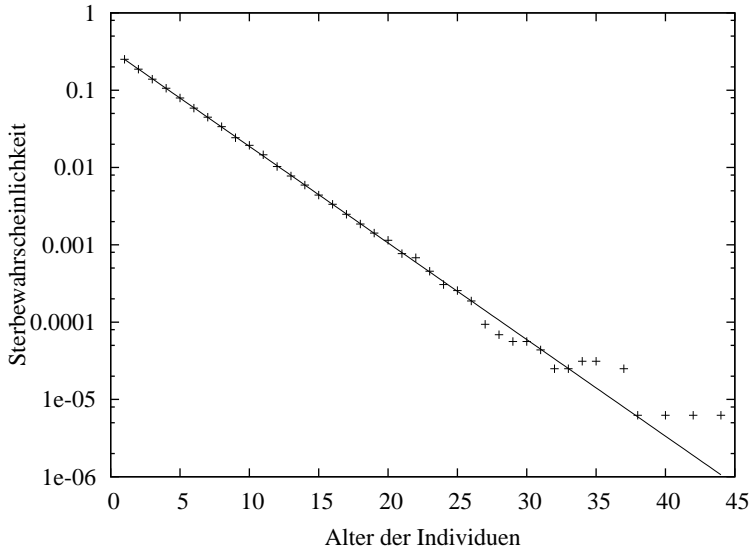
Bevor in den nächsten Abschnitten die einzelnen Leseschemata genauer diskutiert werden, werden zunächst kurz die Ergebnisse einer *zufälligen* Simulation vorgestellt.

5.2 Typische Ergebnisse des Zufallsmodells

Die Simulationen des *Zufallsmodells* werden im weiteren als Referenzsystem verwendet. Statt wie in der Wechselwirkungsphase üblich, die Matrix $A(g, g')$ zu verwenden, um die

Fitness jedes Bit-Strings zu ermitteln (siehe hierzu Abschnitt 3.4.1), werden im Zufallsmodell die Bewertungen der wechselwirkenden Individuen um zufällige Werte aus dem Intervall $[-1, 1]$ verändert. In diesem Fall hängt die Fitness der Bit-Strings *nicht* von ih-

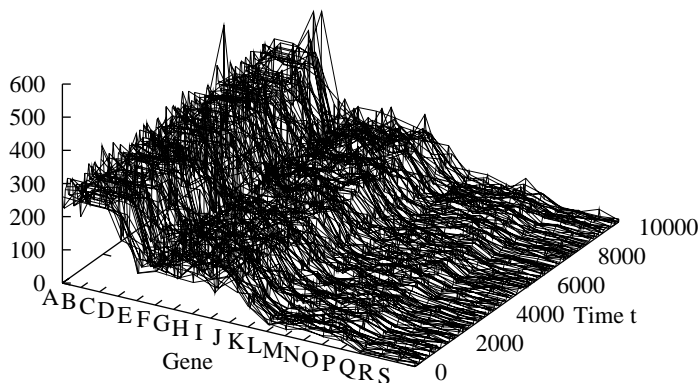
Abbildung 5.1: Sterbewahrscheinlichkeiten im Zufallsmodell



Diese Abbildung zeigt die Sterbewahrscheinlichkeiten für das Zufallsmodell, zur Selektion wurde das Sortiermodell verwendet (siehe Abschnitt 3.4.2). Die Rhomben stammen aus einer Simulation von 64 Bit-Strings über eine Laufzeit von $t_{sim} = 10\,000$ Zeitschritten. Die durchgezogene Linie entspricht der theoretischen Sterbewahrscheinlichkeit für $p = 0.25$, wie sie in Anhang A in Gleichung (A.1) angesetzt wurde. Man erkennt, daß die Kurve die Daten sehr gut beschreibt. Die Abweichungen ab dem Alter $t = 25$ sind eine Folge der niedrigen Statistik.

rem genetischen Inhalt ab, so daß im Prinzip alle Individuen gleich sind. Bei Verwendung des Sortiermodells (siehe Abschnitt 3.4.2) wird daher jeder Bit-String mit derselben Eliminationswahrscheinlichkeit p entfernt. Dies führt zu einer einfachen Altersverteilung, wie die Rechnungen in Anhang A zeigen. In Abbildung 5.1 werden die Daten aus einer Simulation von 64 Bit-Strings mit je 32 Genen über eine Laufzeit von $t_{sim} = 10\,000$ Zeitschritten mit der theoretischen Kurve nach Gleichung (A.1) für eine Eliminationswahrscheinlichkeit von $p = 0.25$ verglichen. Man erkennt, daß die Kurve die Daten sehr gut beschreibt. Die etwas stärkeren Abweichungen ab dem Alter $t = 25$ sind eine Folge der niedrigen Statistik bei hohem Sterbealter.

Abbildung 5.2: Genbesetzungszahlen im Zufallsmodell



Diese Abbildung zeigt die Genbesetzungszahlen für das Zufallsmodell (siehe Text). Die Zahlen ergeben sich durch die Summation über die gesamte Bevölkerung. Die Gene sind bis auf Fluktuationen entsprechend ihrer Länge l mit der Wahrscheinlichkeit 2^{-l} vertreten. Die dazugehörige Altersverteilung sieht man in Abbildung 5.1.

Da im Zufallsmodell einzelne Gene für einen Bit-String keinen selektiven Vorteil haben, kann man sehr leicht die durchschnittlichen Genbesetzungszahlen abschätzen. Ein Gen i , das die Länge l_i Bits hat, sollte mit der Wahrscheinlichkeit $P_g = 2^{-l_i}$ vorhanden sein.

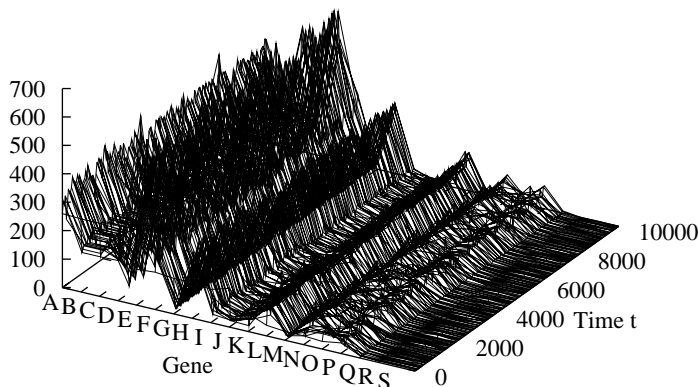
Damit sind längere Gene gegenüber kürzeren Genen unterdrückt. Wegen der Eliminationswahrscheinlichkeit von $p = 0.25$ ist mit hohen Fluktuationen zu rechnen. Dieses bestätigt sich, wie man aus Abbildung 5.2 ersehen kann.

5.3 Höchste Packungsdichte

5.3.1 Ergebnisse von Simulationen

Betrachten wir nun im Gegensatz zu den Ergebnissen des Zufallsmodells die Genbesetzungszahlen, wenn die Wechselwirkung der Gene über die Wechselwirkungsmatrix stattfindet. Abbildung 5.3 zeigt, daß hier ganz andere Verhältnisse gelten. Jeder Bit-String nimmt pro Zeitschritt im Mittel an $\alpha = 10$ Wechselwirkungen teil. Nach etwa 500 Zeitschritten bildet sich aus derselben Anfangspopulation, die zu Abbildung 5.2 geführt hat, ein neuer Gleichgewichtszustand heraus, in dem die Gene zwar nicht entsprechend ihrer Länge in Bits auftreten, in dem sich aber ebenfalls konstante Proportionen der Genbesetzungszahlen (bis auf Fluktuationen) herausbilden.

Abbildung 5.3: Genbesetzungszahlen bei höchster Packungsdichte



Diese Abbildung zeigt die Genbesetzungszahlen für einen Lauf, in dem die in Tabelle 5.1 gezeigte Wechselwirkungsmatrix $A(g, g')$ verwendet wurde. Die Bevölkerungsgröße betrug 64 Individuen mit je 32 Bits. Die Anfangspopulation ist identisch zu der aus Abbildung 5.2. Die Ergebnisse werden im Text diskutiert.

Die gleich langen Gene A bis D (vergleiche hierzu Tabelle 3.1) sind in unterschiedlichen Zahlen vertreten: Gen D ist unterdrückt, während Gen A deutlich überhöht vorliegt. Die höchsten Besetzungszahlen liegen aber für Gen E vor, das ein Bit länger als die Gene A bis D ist. Dagegen ist Gen G, das wie Gen E aus vier Bits besteht, kaum noch vorhanden. Ähnliches erkennt man für die ebenfalls gleich langen Gene L bis P: Gen L ist deutlich überhöht, während Gen M und O unterdrückt sind. Hierfür muß die Selektion in Verbindung mit der Wechselwirkungsmatrix verantwortlich sein. Deshalb untersuchen wir einmal genau die verwendete Wechselwirkungsmatrix, die in Tabelle 5.1 dargestellt ist.

5.3.2 Erklärung der Ergebnisse

Gen E ist in jedem Bit-String etwa 7 mal vorhanden (in dem Beispiel aus Abbildung 5.4 kommt es nur 5 mal vor.), damit findet mit relativ hoher Wahrscheinlichkeit eine Wechselwirkung von Gen E mit sich selbst statt. Aus Tabelle 5.1 kann man aber ersehen, daß dies eine sehr ungünstige Wechselwirkung ist. Die Bewertung der wechselwirkenden Individuen

Tabelle 5.1: Eine zweite Wechselwirkungsmatrix $A(g, g')$

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	0.864	0.819	0.712	-0.678	0.844	0.377	-0.778	0.929	-0.517	-0.958	-0.832	0.747	0.384	0.605	0.692	0.109	-0.936	-0.701	-0.999	0.893
B	-0.951	0.731	-0.831	-0.671	0.784	0.527	-0.477	0.724	-0.855	-0.197	0.710	0.383	-0.327	-0.372	0.506	-0.239	0.760	-0.668	0.112	-0.529
C	-0.271	0.379	-0.536	0.666	0.927	0.743	-0.366	-0.976	0.460	-0.730	0.617	0.573	-0.853	0.587	0.589	0.743	-0.978	0.736	-0.424	-0.472
D	-0.753	-0.898	-0.980	-0.584	0.957	0.534	-0.461	-0.458	-0.286	0.530	0.705	-0.458	-0.655	-0.851	0.917	-0.866	0.868	0.648	-0.459	-0.890
E	-0.554	-0.920	0.833	0.824	-0.934	0.881	-0.539	0.837	0.351	-0.778	0.947	-0.812	-0.940	0.586	-0.427	0.452	0.924	0.731	0.372	0.922
F	0.654	0.940	0.518	0.903	-0.580	-0.577	-0.605	-0.219	-0.691	0.191	-0.918	-0.275	-0.035	-0.318	-0.690	-0.699	0.943	-0.873	0.513	-0.642
G	0.240	-0.424	-0.917	-0.862	-0.466	-0.282	0.036	-0.680	0.837	-0.316	-0.713	-0.433	0.747	-0.702	-0.944	-0.204	0.078	-0.534	0.602	0.921
H	0.300	0.377	-0.378	0.015	0.758	0.033	-0.379	-0.701	-0.972	-0.205	-0.211	-0.812	-0.602	-0.650	-0.839	-0.069	0.934	0.920	-0.905	-0.863
I	0.974	-0.172	-0.374	0.188	-0.465	-0.781	-0.258	-0.831	0.581	0.215	0.513	0.371	0.727	-0.529	0.238	0.195	0.097	-0.430	-0.642	-0.439
J	-0.437	-0.332	0.027	-0.532	-0.019	0.882	-0.307	-0.074	-0.739	0.133	-0.903	0.409	-0.536	-0.160	-0.347	-0.840	-0.722	-0.005	0.557	-0.322
K	0.737	0.874	-0.605	-0.575	-0.604	-0.517	-0.443	-0.025	0.001	0.354	0.296	0.196	0.718	-0.359	0.425	0.764	0.168	-0.621	0.109	-0.518
L	-0.372	-0.618	0.335	0.417	-0.062	-0.248	0.066	0.168	0.034	0.259	-0.669	0.918	0.052	0.203	0.336	-0.912	-0.040	-0.554	0.616	0.275
M	-0.350	-0.631	0.924	0.650	0.943	-0.273	0.615	0.891	0.678	0.033	0.163	-0.153	0.507	-0.889	0.033	-0.225	0.572	0.090	-0.586	-0.024
N	0.087	-0.158	-0.479	-0.421	0.520	-0.265	-0.748	0.834	0.220	0.635	0.218	-0.688	-0.616	-0.158	-0.845	-0.747	0.994	0.962	-0.660	-0.099
O	0.281	-0.259	-0.766	-0.809	0.174	-0.212	0.228	-0.686	0.994	-0.836	-0.345	-0.360	-0.403	-0.237	-0.665	0.207	-0.526	0.678	-0.380	0.548
P	0.611	-0.870	0.825	-0.676	-0.537	0.137	0.447	0.131	-0.941	-0.915	-0.778	-0.897	-0.626	-0.288	-0.847	-0.747	-0.820	-0.033	0.478	0.698
Q	0.084	-0.350	0.827	0.777	0.299	-0.530	-0.715	0.586	-0.707	-0.824	-0.386	-0.886	-0.909	0.155	-0.015	0.933	-0.336	-0.827	-0.561	-0.817
R	0.393	-0.196	-0.665	0.237	0.941	-0.624	-0.426	-0.326	0.099	-0.464	-0.455	-0.382	0.551	-0.934	-0.734	-0.070	0.506	-0.187	-0.917	0.763
S	-0.828	-0.861	-0.059	-0.231	-0.849	0.822	0.522	-0.354	0.248	0.610	-0.927	-0.204	-0.283	-0.644	0.846	-0.898	-0.123	0.475	0.091	0.678
T	0.249	0.907	0.648	-0.422	0.798	-0.970	-0.731	-0.360	0.390	-0.852	-0.386	-0.630	0.787	-0.330	0.398	-0.370	-0.617	0.634	-0.764	-0.731

Abbildung 5.4: Individuum mit höchster Packungsdichte

Bitsequenz
0 1 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 1 1 1 1 0 0 0 1 0 0 1 1 0 0
Genmuster
FQAGIFHELAFHEEEEOABDCHLAFKABCHLA

Diese Abbildung zeigt einen Bit-String aus der Simulation, die zu Abbildung 5.3 geführt hat. Jedes Gen steht unter dem Bit, an dem es anfängt. Der Bit-String ist zirkular geschlossen. Die Dekodierung wird in Tabelle 5.2 an einer Teilsequenz genau erläutert.

Es handelt sich hierbei um ein langlebiges Individuum, das 31 Zeitschritte in der Simulation geblieben ist. Da es nur 4 Nachkommen gezeugt hat, hat es die meiste Zeit in der mittleren Hälfte existiert, die zwar überlebt, aber keinen Nachwuchs bekommt.

wird um -0.934 verschlechtert. Dagegen hat das Gen G eine neutrale (0.036) Selbstwechselwirkung. Betrachten wir zunächst noch einmal die Diagonaleinträge der Matrix. Gen A und B unterstützen sich selbst, während Gen C und D sich selbst unterdrücken. Man sieht aber in Abbildung 5.3, daß Gen B deutlich geringere Besetzungszahlen als Gen A und genauso große Besetzungszahlen wie Gen C hat (die Werte von Gen B und C sind identisch).

Tabelle 5.2: Teilsequenz mit höchster Packungsdichte

Bits	Gene
0 0 0 1 0 0	L
0 0 1	A
0 1 0 0	F

Diese Tabelle zeigt die Dekodierung einer Teilsequenz, wenn an jeder Position des Bit-Strings ein Gen anfängt. Liegt in einem Bit-String die Sequenz '000100' vor, so wird dies immer als Gensequenz 'LAF' dekodiert. Diesem folgt ein Gen, das mit '100' beginnt (dies sind die Gene H, K und P-T, wie man aus Tabelle 3.1 ersehen kann). (Entnommen aus [42])

Betrachtet man nur einzelne Gene und ihre Wechselwirkung untereinander, dann läßt sich die in Abbildung 5.3 gezeigte Verteilung nicht verstehen. Deshalb schauen wir uns die Dekodierung einmal etwas genauer an. Wie man aus Tabelle 5.2 erkennen kann, wird z.B. das Gen L nicht allein dekodiert, es tritt immer in Verbindung mit den Genen A und F auf und dahinter steht eines der Gene H, K oder P bis T. (In dem Bit-String aus Abbildung 5.4 kommt diese Sequenz dreimal vor.) Gleiches findet man für alle längeren Gene angefangen bei Gen J, das in einer festen Zweiergruppe auftritt.

Die überlappenden Gene sind in Tabelle 5.3 zusammengefaßt. Hinter der festen Gengruppe

Tabelle 5.3: Überlappende Gengruppen

Gengruppe	Bits	Gengruppe	Bits
JB+C,D	1011	PAF+H,K,P-T	100100
KAB+C,D	10111	QAGIF+H,K,P-T	10010100
LAF+H,K,P-T	000100	RAGIG+I,J	10010101
MAG+I,J	000101	SAGJBC+H-K,P-T	10010110
NABC+H-K,P-T	000110	TAGJBD+C,D	10010111
OABD+C,D	000111		

Diese Tabelle zeigt die überlappenden Gengruppen, wenn bei der Dekodierung eines Bit-Strings an jeder Position ein Gen anfängt. Hinter dem Pluszeichen sind die Gene angegeben, die an der nächsten Stelle folgen können. Man erkennt, daß sehr starke Korrelationen vorliegen.

(z.B. LAF), die aus bis zu 6 Genen bestehen kann, ist noch angegeben, welche Gene direkt dahinter noch möglich sind. Man erkennt, daß sehr starke Korrelationen vorliegen. Gen A liegt automatisch hinter den Genen K bis T vor. Als weiterer Punkt fällt auf, daß sowohl hinter Gen B als auch hinter Gen D nur Gen C oder Gen D vorliegen kann. Dies hat zur Folge, daß sich die Gengruppen BD_iC ergeben, wobei i die Werte $0, 1, 2, \dots$ annehmen kann. (In Abbildung 5.4 findet man je ein Beispiel für $i = 0$ und $i = 1$.) Damit ergibt sich, daß jedes Gen B mit genau einem Gen C korreliert ist, so daß geklärt ist, warum sie mit identischen Zahlen vorliegen. Die Gene D und E können als einzige Gene in einer längeren Sequenz vorliegen, die nur aus einem Gentyp besteht, (in dem Beispiel aus Abbildung 5.4 stehen 4 Gene E hintereinander,) da sie aus nur Einsen bzw. nur Nullen bestehen (siehe Tabelle 3.1). Gen D ist dabei von den Genen B und C eingeschlossen.

Wie man deutlich erkennt, genügt es nicht, die Wechselwirkung einzelner Gene zu betrachten, da sie nicht isoliert auftreten können, sondern man muß die Wechselwirkung von ganzen Gengruppen untersuchen. In der Literatur taucht in diesem Zusammenhang der Begriff *Hitchhiker-Effekt* auf [25, 89, 93, 113]. Dies wird in Tabelle 5.4 anhand der Gene M und N vorgeführt, die zu Beginn gleiche Besetzungszahlen haben. Betrachtet man nur den oberen Teil, in dem die reduzierte Wechselwirkungsmatrix dargestellt ist, die nur die Gene M und N enthält, so erkennt man, daß Gen M eine bessere Bewertung als Gen N erhält. Also sollte die Besetzungszahl von M relativ zu N ansteigen. Berücksichtigt man aber die Korrelationen und berechnet die mittlere Wechselwirkung, die die Gruppen MAG und NABC aufeinander ausüben (siehe der untere Teil von Tabelle 5.4), so erkennt man, daß sich hier die Verhältnisse *umgekehrt* haben, da nun Gen N die bessere Bewertung relativ zu Gen M hat. Damit läßt sich erst mittels der Korrelationen verstehen, warum Gen M eine niedrigere Besetzungszahl (etwa 15) als Gen N (etwa 60) hat (siehe Abbildung 5.3).

Tabelle 5.4: Vergleich von Einzel- und Gruppenwechselwirkungen

	M	N
M	0.505	-0.889
N	-0.616	-0.158
	MAG	NABC
MAG	0.252	0.021
NABC	-0.334	-0.020

Diese Tabellen zeigen die reduzierten Wechselwirkungsmatrizen zwischen den Genen M und N, die obere ohne Berücksichtigung der Korrelationen und die untere mit. In der unteren Tabelle sind die Werte auf eine Wechselwirkung normiert. Erst die Korrelationen lassen verstehen, warum Gen M eine niedrigere Besetzungszahl (etwa 15) als Gen N (etwa 60) hat.

Die bis hierhin durchgeführten Überlegungen sind natürlich nur eine erste Näherung, die aber schon recht gute Einblicke in die Dynamik liefert. Die Korrelationen, die in den Tabellen 5.3 und 5.4 gezeigt werden, sind die festen, garantierten Korrelationen. Es gibt aber noch abgeschwächte Korrelationen. Betrachten wir z.B. noch einmal die drei

Gengruppen LAF, die in Abbildung 5.4 vorliegen. Vor dem Gen L ist entweder direkt das Gen H oder eine Gengruppe der Form HE_i mit $i = 1, 2, \dots$, so daß sich effektiv noch größere Gengruppen als die in Abbildung 5.3 ergeben. Analoges ist schon weiter oben für die drei Gene B, C und D erklärt worden.

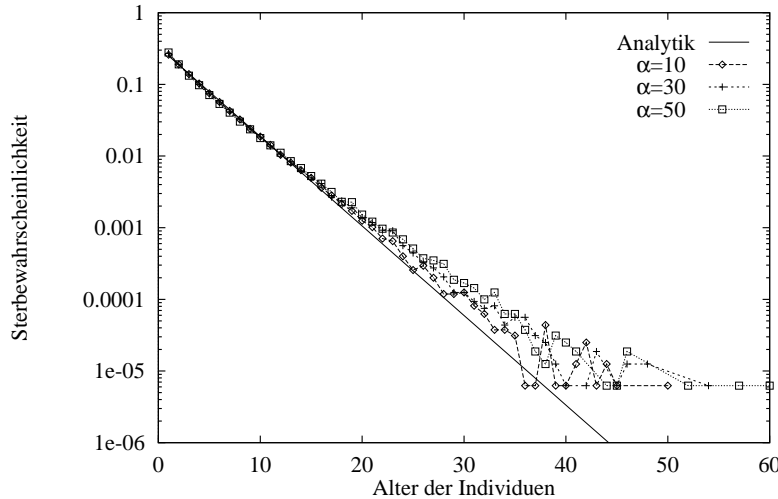
Jeder Bit-String enthält 32 Gene. Bei einer Wechselwirkungsrate von $\alpha = 10$, wie sie bei der Erstellung von Abbildung 5.3 verwendet wurde, nimmt im Mittel nur jedes dritte Gen eines Bit-Strings an einer Wechselwirkung teil. Dies führt zu einer stark verrauschten Bewertung. Da in einem Bit-String normalerweise nur etwa 10 verschiedene Gene enthalten sind, kann man aber davon ausgehen, daß jede Genart etwa einmal bewertet wurde. Dieses scheint schon auszureichen, da Simulationen mit höheren Wechselwirkungsraten ($\alpha = 30$ und 50) qualitativ zu denselben Ergebnissen geführt haben: Die Gene A und E dominieren die Simulation.

Qualitativ gleiches Verhalten zeigt sich auch für andere Anfangspopulationen.

5.3.3 Der Einfluß der Selektion

Andere Wechselwirkungsmatrizen ermöglichen zwar andere dominierende Gene, aber die Tendenz bleibt erhalten: Innerhalb von 500 bis 1000 Zeitschritten wird ein stationärer Zustand erreicht, in dem sich (bis auf Fluktuationen) konstante Proportionen der Genbesetzungszahlen ausgebildet haben.

Abbildung 5.5: Vergleich verschiedener Wechselwirkungsraten



Diese Abbildung vergleicht die Altersverteilungen für verschiedene Wechselwirkungsraten α . Die Daten stammen jeweils aus Simulationen von 64 Bit-Strings über eine Laufzeit von $t_{sim} = 10\,000$ Zeitschritten. Die durchgezogene Linie entspricht der theoretischen Sterbewahrscheinlichkeit für $p = 0.25$, wie sie in Anhang A in Gleichung (A.1) angesetzt wurde. Man erkennt, daß deutliche Abweichungen vorliegen, die mit ansteigender Wechselwirkungsrate stärker werden.

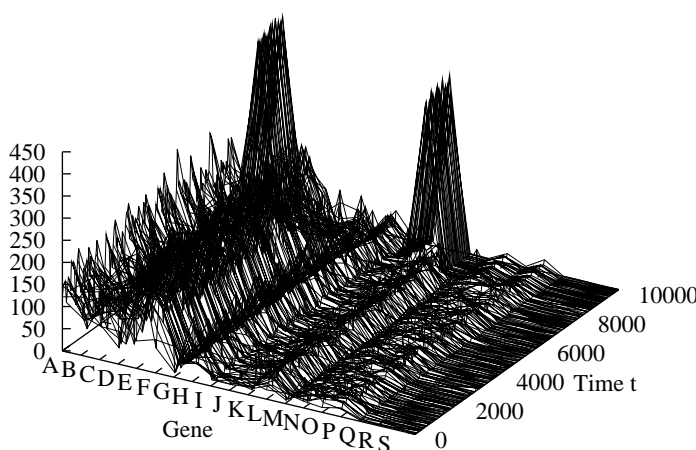
Obwohl dieses Modell keine ständig fortschreitende Evolution zeigt, ist es doch interessant, weil es erlaubt, den Effekt der Selektion genauer unter die Lupe zu nehmen. Man sieht einen besonders starken Einfluß auf die Altersverteilung der Individuen. Dies soll anhand von Abbildung 5.5 erläutert werden: Für ein Zufallsmodell erwartet man eine Exponentialverteilung (siehe Abbildung 5.1 und Anhang A, bzw. die durchgezogene Linie in Abbildung 5.5). Abbildung 5.5 zeigt die gemessene Altersverteilungen für verschiedene Wechselwirkungsraten α und vergleicht diese mit der theoretischen Kurve. Man erkennt sehr deutliche Abweichungen für hohes Individuenalter, die mit steigender Wechselwirkungsrate stärker ausgeprägt sind. Dies liegt daran, daß Individuen mit einem günstigen Geninhalt in der Wechselwirkungsphase eine hohe Bewertung erhalten und daher bei Verwendung des Sortiermodells in das beste Viertel einsortiert werden. Das beste Viertel

überlebt *und* zeugt Nachwuchs. Bei den Simulationen, die zu Abbildung 5.5 geführt haben, hat immer genau eine Punktmutation stattgefunden. Eine genaue Analyse zeigt, daß eine Punktmutation im Mittel 4 Gene betrifft. Daher entstehen mit hoher Wahrscheinlichkeit ungünstigere Individuen, die schneller eliminiert werden, so daß sich ebenfalls eine Überhöhung bei niedrigem Individuenalter ergibt. Damit erhält man bei mittlerem Alter (etwa 4 bis 12 Zeitschritte) eine *Delle* in der Verteilung.

5.3.4 Geringere Korrelationen

Die Korrelationen lassen sich deutlich verringern, wenn nicht an jedem Bit ein Gen anfängt, sondern z.B. nur an jedem zweiten Bit. Damit werden die Gruppen korrelierter Gene kleiner. Aus KAB wird KB. Die längsten Gruppen werden von 6 auf 3 Gene reduziert (vergleiche Tabelle 5.3), gleichzeitig reduziert sich die Zahl der Gene im Bit-String von 32 auf 16. Nun sind die Korrelationen gering genug, daß sich eine einzelne Gruppe von Genen als die fitteste Gruppe herausbilden kann. Abbildung 5.6 zeigt die Genbesetzungszahlen für eine Simulation von 64 Bit-Strings der Länge 32 Bits, bei denen an jedem zweiten Bit ein Gen anfängt. Die Simulationszeit betrug $t_{sim} = 10\,000$ Zeitschritte. Nach etwa 9 000 Zeitschritten stellt sich ein Zustand ein, in dem die Gengruppe KB dominiert (ein Bit-String enthält diese Gruppe 8 mal). Wie Simulationen mit unterschiedlichen Anfangspopulationen zeigen, scheint es sich bei der Genkombination KB um einen globalen Attraktor zu handeln. Nur der relativ hohen Mutationsrate von 1 Punktmutation pro Replikation (dies entspricht einer Mutationsrate von etwa 3% pro Bit) ist es zu verdanken, daß auch noch andere Gene (z.B. A und H) mit relativ hoher Besetzungszahl existieren.

Abbildung 5.6: Genbesetzungszahlen bei niedrigerer Packungsdichte



Diese Abbildung zeigt die Genbesetzungszahlen für einen Lauf, in dem die in Tabelle 5.1 gezeigte Wechselwirkungsmatrix $A(g, g')$ verwendet wurde. Die Bevölkerungsgröße betrug 64 Individuen mit je 32 Bits. Es fängt im Gegensatz zur Simulation aus Abbildung 5.3 nur an jedem zweiten Bit ein Gen an, damit enthält jedes Individuum nur 16 statt 32 Gene. Die Anfangspopulation ist identisch zu der aus den Abbildungen 5.2 und 5.3. Die Ergebnisse werden im Text diskutiert.

Wie die in diesem Abschnitt diskutierten Simulationen zeigen, ist dieses Dekodierschema, das wegen der starken Korrelationen relativ schnell zu einem Gleichgewichtszustand führt, als Evolutionsmodell nicht sehr geeignet. Im nächsten Abschnitt wird deshalb ein Dekodierschema untersucht, das keine Korrelationen zwischen den Genen aufweist.

5.4 Das Dekodierschema *Gen an Gen*

5.4.1 Die Dekodierung

Das zweite Leseschema vermeidet Korrelationen zwischen einzelnen Genen, indem ein zweites Gen erst anfängt, *nachdem* das erste vollständig dekodiert ist. Abbildung 5.7 zeigt ein Beispiel für eine solche Dekodierung. Es kann dabei vorkommen, daß die letzten Bits kein Gen mehr dekodieren, dieser Abschnitt wird als *Junk* bezeichnet.

Abbildung 5.7: Dekodierung ohne Überlapp

C	S	F	D	G	N	A	-
1 1 0	1 0 0 1 0	1 1 0	0 1 0 0	1 1 1	0 1 0 1	0 0 0 1 1 0	0 0 1 0
↓							
1 0 0	1 0 0	1 0 1 1	0 0 1	0 0 1	1 1 0	1 0 1 0	0 0 1 1 0 0 1 0
P	J	A	A	C	I	A	H - -

Diese Abbildung zeigt einen Bit-String, bei dem keine überlappenden Gene existieren. Der Einfluß einer Punktmutation (durch den Pfeil gekennzeichnet) kann dabei sehr drastisch sein, wie man am Vergleich der beiden Dekodierungen ablesen kann. (Entnommen aus [42])

5.4.2 Der Einfluß von Punktmutationen

Das Schema *Gen an Gen* vermeidet überlappende Gene. Damit sind keine Korrelationen vorhanden, die in der Wechselwirkungsphase Bedeutung haben.

Untersucht man den Einfluß von Punktmutationen, so erkennt man, daß aber doch irgendwelche Korrelationen vorhanden sind: Eine genaue Analyse zeigt, daß bei der Verwendung der höchsten Packungsdichte eine Punktmutation im Mittel 4 Gene verändert. Da jeder Bit-String 32 Gene enthält, entspricht dies 1/8 des genetischen Inhaltes. Beim Dekodierschema *Gen an Gen* können wesentlich stärkere Veränderungen auftreten. In Abbildung 5.7 sieht man ein Beispiel, in dem eine Punktmutation am zweiten Bit den Leserahmen vollständig verschiebt. Im Prinzip sind alle Gene davon betroffen. Aus den ursprünglichen 7 Genen werden nun 8 Gene, allerdings ist das Gen A und das Gen C in beiden Bit-Strings vorhanden. Da bei der Wechselwirkung die Reihenfolge der Gene keine Rolle spielt, bewirkt diese Punktmutation, daß aus den 5 Genen DFGNS im *Vorfahren* die 6 Gene AAHIJP im *Nachkommen* werden und daß die 2 Gene A und C erhalten bleiben.

Eine genaue Untersuchung der 32 möglichen Punktmutationen, die am Bit-String aus Abbildung 5.7 durchgeführt werden können, ergibt, daß 16 dieser Punktmutation den Leserahmen so verschieben, daß mindestens zwei Gene im Vorfahren davon betroffen sind. Ähnliche Verhältnisse gelten bei allen Bit-Strings. Auf diese Weise erzeugt eine Punktmutation mit hoher Wahrscheinlichkeit einen großen Sprung im Raum aller möglichen Individuen. In dieser Arbeit wird Wert darauf gelegt, daß eine Punktmutation einer geringen Veränderung (d.h. einem geringen Rauschniveau) entspricht, so daß eine Punktmutation ein lokaler Operator ist. Daher muß man sich einen Ausweg überlegen, der den Mutationsoperator lokal macht.

Bei den Simulationen, die in diesem Abschnitt vorgestellt werden, wurde als Quelle der Veränderung exakt eine Punktmutation ausgewählt. Diese sollte, damit der Mutationsoperator möglichst lokal wirkt, nur ein Gen verändern. Das läßt sich erreichen, indem man das betroffene Gen *herausschneidet* und an das Ende des Bit-Strings verschiebt (siehe Abbildung 5.8).

Abbildung 5.8: Herausschneiden des mutierten Gens

Vor der Mutation am 2. Bit																
C	S			F		D	G		N		A	-				
1 1 0	1	0 0 1	0	1 1	0	0 1	0 0	1 1	1	0 1	0 1	0 0 0	1 1 0	0 0	1	0
1 0 0	1	0 1 1	0	0 1	0 0	1	1 1	0 1	0	1	0 0	1 1 0	0 0 1	0 1	0	0
S			F		D	G		N		A	F					

Nach der Mutation und dem Verschieben des 1. Gens

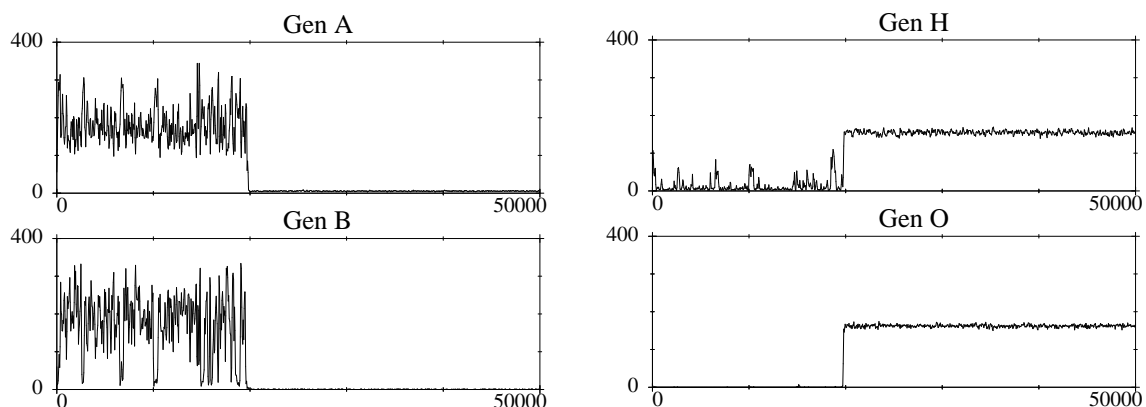
Diese Abbildung zeigt einen Bit-String, bei dem keine überlappenden Gene existieren. Der Einfluß einer Punktmutation am zweiten Bit wird demonstriert. Das veränderte Gen wird *herausgeschnitten* und an das Ende des Bit-Strings geschoben. Das mutierte Bit wird in beiden Fällen fett gedruckt.

Vergleicht man die beiden Bit-Strings Bit für Bit, so unterscheiden sie sich an 18 der 32 Positionen. Untersucht man aber die Genebene, so erkennt man, daß der Leserahmen kaum verändert wird, so daß nur das Gen C in das Gen G mutiert wird. Dies entspricht einer geringen Änderung des Geninhaltes.

Dieser veränderte Mutationsoperator läßt sich biologisch nur sehr schwer motivieren. Er widerspricht auch dem Grundsatz, möglichst einfache Regeln zu verwenden, der dieser Arbeit zugrunde liegt. Dennoch ist davon Gebrauch gemacht worden, da man sehr interessante Ergebnisse erhält, die im Folgenden vorgestellt werden.

5.4.3 Simulationsergebnisse

Bei den Simulationen, die in Abschnitt 5.3.1 vorgestellt wurden, kamen alle Gene vor. Bei den Simulationen aus Abschnitt 5.3.4 wurden die meisten Gene aus der Population eliminiert, sobald der stabile Zustand mit den Genen B und K erreicht wurde. Es hat aber kein Wechsel zwischen verschiedenen (meta-)stabilen Zuständen stattgefunden, so daß man die Dynamik, die in Abschnitt 5.3 beschrieben wurde, als reinen Optimierungsprozeß auffassen kann. Die Ergebnisse, die mit dem Dekodierschema *Gen an Gen* erzielt werden können, sehen dagegen qualitativ anders aus [42].

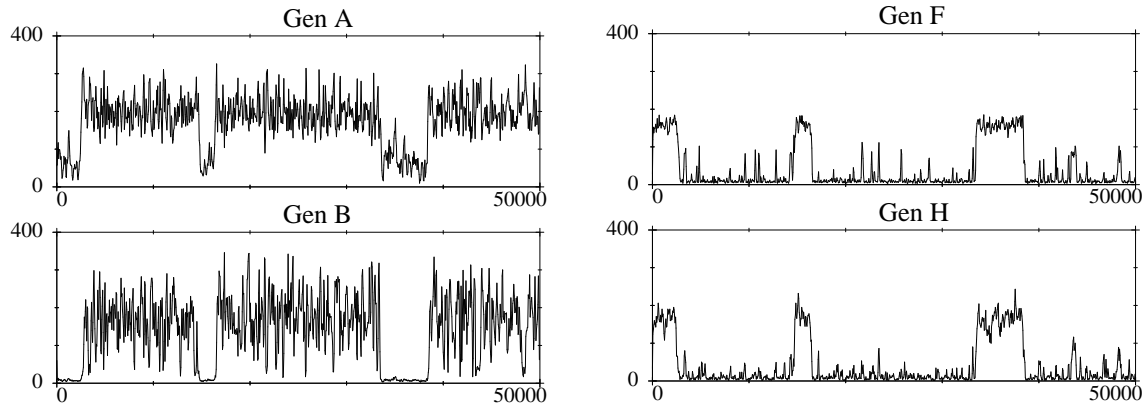
Abbildung 5.9: Beispiel für *genetic takeover*

Diese Abbildung zeigt die Genbesetzungszahlen für die vier erfolgreichsten Gene. Die Zahlen ergeben sich durch die Summation über die gesamte Bevölkerung. Zu Beginn dominieren die Gene A und B für fast 20000 Zeitschritte, bevor dann die Gene H und O aufkommen und übernehmen. Gene mit geringen Besetzungszahlen werden nicht gezeigt. Zum Beispiel liegt außer den Genen H und O auch noch das Gen N vor, dessen Besetzungszahl liegt aber unter 50. Normalerweise haben etwa 12 der 20 Gene vernachlässigbare Besetzungszahlen. Die Wechselwirkungsrate α beträgt 20. (Entnommen aus [42])

Während der Simulation kann ein Gen lange Zeit nicht vorhanden sein, das später wieder auftaucht und sogar die Simulation dominiert. Dies passiert mit Gen O in Abbildung 5.9. Auf diese Weise ist eine sich ständig ändernde Population möglich. Hieraus folgt, daß keine

frequency-independent fitness [91] existiert. Man kann also keine Fitnessfunktion global angeben, die extern ausgewertet werden kann. Man beobachtet mehrere Phänomene, die sich auf den Einfluß der Wechselwirkungsmatrix zurückführen lassen:

Abbildung 5.10: Beispiel für Bistabilität



Diese Abbildung ist analog zu Abbildung 5.9. Es wurde dieselbe Anfangspopulation verwendet. Im Unterschied zu Abbildung 5.9 beträgt die Wechselwirkungsrate α hier nur 10. Man erkennt, daß sich hier zwei stabile Phasen abwechseln. Die erste besteht aus der Dominanz der Gene A und B, wie in Abbildung 5.9, während die zweite von den Genen F und H dominiert wird. (Entnommen aus [42])

- Es kann Kooperation zwischen zwei Genen geben, wie man an den Genen H und O in Abbildung 5.9 erkennen kann.
- Zwei Gene können sich gegenseitig unterdrücken, wie man an den Genen B und F in Abbildung 5.10 erkennen kann.

Die Wechselwirkungsmatrix, die diesen Simulationen zugrunde liegt, ist in Kapitel 3 in Tabelle 3.2 dargestellt. Das charakteristische Verhalten des simulierten Systems – *Bistabilität* (siehe Abbildung 5.10) und *genetic takeover* (Übernahme durch eine andere Art, siehe Abbildung 5.9) – ist unabhängig von der Anfangspopulation und auch weitgehend unabhängig von der gewählten Wechselwirkungsmatrix. Diese entscheidet nur, wieviele und welche stationären Zustände möglich sind. Einen starken Einfluß auf das Verhalten des Systems hat die Wechselwirkungsrate α , die in Abschnitt 3.4.1 definiert wurde.

Die Simulationen, die zu den Abbildungen 5.9 und 5.10 geführt haben, wurden mit derselben Anfangspopulation von 64 Bit-Strings, die jeweils 32 Bits enthalten, durchgeführt. Bei Abbildung 5.9 beträgt $\alpha = 20$ und bei Abbildung 5.10 ist $\alpha = 10$. Jeder Bit-String enthält etwa 6 bis 10 Gene, damit entspricht $\alpha = 10$ etwa 1 bis 1.5 Wechselwirkungen pro Gen, während $\alpha = 20$ schon 2 bis 3 Wechselwirkungen entspricht. Da die wechselwirkenden Gene zufällig ausgewählt werden (siehe Abschnitt 3.4.1), ist hierbei mit starken Schwankungen zu rechnen.

Bei einer höheren Wechselwirkungsrate nimmt jedes Gen eines Bit-Strings mit höherer Wahrscheinlichkeit an mindestens einer Wechselwirkung teil. Dieses führt zu geringeren Schwankungen in der berechneten Bewertung. Bit-Strings, die in einem Zeitschritt erfolgreich waren, sind dies mit hoher Wahrscheinlichkeit auch im nächsten Zeitschritt. Es bildet sich also sehr schnell eine ziemlich homogene Bevölkerungsstruktur aus, die zu langen (meta-)stabilen Phasen führt, die nur selten durch eine andere Phase ersetzt werden (siehe Abbildung 5.9). Dieses Verhalten wird in der Literatur als *Punctuated Equilibria* [31, 85] bezeichnet.

Bei einer niedrigeren Wechselwirkungsrate unterliegt die Bewertung eines einzelnen Bit-Strings im Laufe seines Lebens starken Schwankungen, selbst wenn die Bevölkerungsstruk-

tur sich nicht verändert. Auf diese Weise können metastabile Zustände leichter zerstört und durch andere Zustände ersetzt werden. Es ist sogar möglich, daß ein Zustand immer wieder auftaucht. In Abbildung 5.10 sieht man, daß sich zwei Phasen häufig abwechseln, die eine wird durch die Gene A und B dominiert, die andere durch die Gene F und H.

5.4.4 Bewertung des Dekodierschemas *Gen an Gen*

In diesem Abschnitt ist das Dekodierschema *Gen an Gen* verwendet worden. Um den Einfluß einer Punktmutation so gering wie möglich zu halten, wurde dabei das mutierte Gen aus dem Bit-String *herausgeschnitten* und an dessen Ende geschoben. Zur Selektion ist das Sortiermodell, bei dem ein Viertel der Population eliminiert wird, verwendet worden. Dieses Modell zeigt ermutigende Resultate, da man wie auch in natürlichen Evolutionsystemen zum einen Stabilität und zum anderen die Möglichkeit zur Veränderung findet. Dieses gilt, obwohl bei der Erstellung des Modells räumliche Homogenität vorausgesetzt wurde, die die Formation von verschiedenen Nischen und damit von weiterer Struktur verhindert.

Allerdings hat dieses Schema auch einige Nachteile, die es nicht geeignet erscheinen lassen, es weiter zu untersuchen. Das Herausschneiden des mutierten Gens läßt sich biologisch nicht motivieren, außerdem ist es bei mehreren Punktmutationen keine praktikable Methode. Da Punktmutationen ohne Herausschneiden den Leserahmen sehr stark verschieben können, ist es sinnvoll, nach einem Dekodierschema zu suchen, daß diesen Nachteil vermeidet. Ein solches wird im nächsten Abschnitt ausführlich diskutiert.

5.5 Verwendung von Startcodons

5.5.1 Die Dekodierung

Das dritte Leseschema verwendet Startcodons, wie sie in der Natur vorgefunden werden [29, 75]. Ein Gen wird im Bit-String erst dann dekodiert, wenn ein solches Startcodon gefunden wurde. In dieser Arbeit wird als Startcodon die Bitsequenz ‘110’ verwendet. In Abbildung 5.11 wird derselbe Bit-String, der schon in Abbildung 5.7 verwendet wurde, untersucht und der Einfluß einer einzelnen Punktmutation dargestellt. Man erkennt, daß normalerweise nur ein Gen von einer Punktmutation betroffen ist. In einzelnen Fällen können zwei Gene verändert werden. In Abbildung 5.11 wäre das der Fall, wenn die Punktmutation im zweiten Startcodon stattfände.

Abbildung 5.11: Individuum mit Startcodons

s	c	S				-	s	c	I	-	-	s	c	A	-					
		s	c	F																
1	1	0	1	0	0	1	1	0	0	1	1	1	0	1	0	0	0	1	0	
↓																				
1	0	0	1	0	0	1	1	0	0	1	1	1	0	1	0	0	0	1	0	0
		-	-	-	-	-	s	c	F		-	s	c	I	-	-	s	c	A	-

Diese Abbildung zeigt einen Bit-String, bei dem Startcodons verwendet werden (durch ‘sc’ gekennzeichnet, siehe auch Tabelle 3.1). Der Einfluß einer Punktmutation (durch den Pfeil gekennzeichnet) ist dabei immer auf maximal zwei Gene beschränkt. Bits, die weder zu einem Startcodon noch zu einem Gen gehören, sind durch ‘-’ gekennzeichnet. Die *Informationslänge* (siehe Text) beträgt vor der Mutation 28 und nach der Mutation 20 Bits. (Entnommen aus [42])

Die Einführung von Startcodons hat zwei positive Aspekte:

1. Startcodons garantieren, daß eine Punktmutation ein lokaler Operator ist. Dies kann als Ursache dafür gesehen werden, daß sie auch in DNS-Sequenzen zu finden sind. Sie sind nämlich ein wirksames Mittel, um die in Abschnitt 2.1 diskutierte *Fehlerkatastrophe* in den Griff zu bekommen.
2. Die Einführung von Startcodons bewirkt, daß Bits existieren, die weder zu einem Startcodon noch zu einem Gen gehören und somit nicht interpretierte Information darstellen, die als *Junk* bezeichnet wird. In Abbildung 5.11 werden sie durch ‘-’ gekennzeichnet. In Junkregionen können sich Veränderungen ansammeln, die erst zu einem späteren Zeitpunkt, wenn ein Startcodon erzeugt wird, zur Geltung kommen.

Damit kann man eine neue Meßgröße einführen, die in Abschnitt 3.5 noch nicht vorgestellt wurde, die **Informationslänge**. Diese bezeichnet die Anzahl der Bits, die entweder in einem Startcodon oder in einem Gen enthalten sind. Bei einer zufälligen Bevölkerung, die aus 96 Bit langen Individuen besteht, beträgt die durchschnittliche Informationslänge 66 Bits, wenn als Startcodon die Sequenz ‘110’ verwendet wird.

Bevor der Einfluß der Startcodons auf die Dynamik diskutiert wird, sollen erst einmal einige theoretische Überlegungen zur grundsätzlichen Beschaffenheit von Startcodons durchgeführt werden.

5.5.2 Der Aufbau von Startcodons

Eine einfache Methode Startcodons zu finden, besteht darin, jedes Bit eines Bit-Strings daraufhin zu untersuchen, ob es ein Startcodon einleitet. Aus den vorigen Abschnitten ist bekannt, daß der Überlapp von Genen zu starken, nicht gewünschten Korrelationen führen kann. Wählt man sehr lange Startcodons von etwa 10 Bits, so besteht nur geringe Gefahr eines Überlapps. Dies hat aber zur Folge, daß nur sehr wenige Gene überhaupt dekodiert werden. Also darf das Startcodon nicht zu lang sein.

In dieser Arbeit wurde eine Länge von 3 Bits für die Startcodons verwendet. Damit nun nicht schon zwei Startcodons überlappen, bietet sich folgendes allgemeines Aussehen für Startcodons an: $0_i 1_j$ bzw. $1_k 0_l$, wobei $i, j, k, l \geq 1$ gilt. Bei einer Länge von maximal drei Bits kommen somit nur die Sequenzen ‘01’, ‘10’, ‘001’, ‘011’, ‘110’ und ‘100’ als mögliche Startcodons in Frage.

Von diesen 6 Startcodons erfüllt für den in dieser Arbeit verwendeten Genbaum (siehe Abbildung 3.2) nur das letzte (‘110’) die Forderung, daß zwei Gene nicht überlappen. Im Extremfall (Startcodon ‘10’ und Gen Q) kann ein einzelnes Gen schon drei Startcodons enthalten. Das Startcodon ‘110’ ist zwar auch in zwei Genen (N und S) vollständig enthalten, da es aber am Ende der Gene steht, überlappen die beiden Gene nicht (siehe Tabelle 3.1 und Abbildung 5.11).

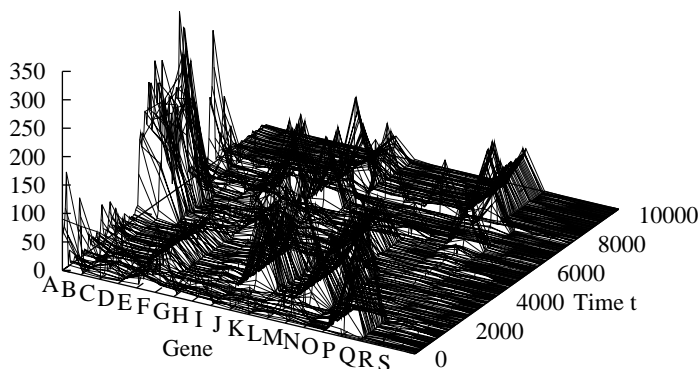
In dieser Arbeit wurde *nur* das Startcodon ‘110’ verwendet.

5.5.3 Erste Ergebnisse

Eine Simulation dieses Modells führt zu ähnlichen Ergebnissen wie die Simulationen mit dem Dekodierschema *Gen an Gen*. Abbildung 5.12 zeigt Simulationsergebnisse für eine Population von 64 Bit-Strings, die je aus 96 Bits bestehen. Diese Länge wurde gewählt,

damit ein Bit-String aus der Anfangspopulation etwa 10 bis 12 Gene enthält, dies entspricht ungefähr der Anzahl von Genen, die ein 32 Bit langes Individuum beim Dekodierschema *Gen an Gen* hat. Wie man erkennen kann, existiert zu Beginn eine Verteilung entsprechend der Länge der Gene, wie im Zufallsmodell. Innerhalb von weniger als 50 Zeitschritten hat sich eine Dominanz der Gene A, E, F und H ausgebildet, die aber schon nach etwa 200 Zeitschritten durch die Dominanz des Gens P abgelöst wird. Bis etwa zum Zeitschritt 8000 finden noch mehrere Wechsel der dominanten Gene statt, bevor dann der sehr stabile Zustand, der die Gene H und O enthält, erreicht wird.

Abbildung 5.12: Genbesetzungszahlen bei Verwendung von Startcodons



Diese Abbildung zeigt die Genbesetzungszahlen bei Verwendung von Startcodons. Die Population besteht aus 64 Individuen mit je 96 Bits. Die Wechselwirkungsrate α beträgt 10.

Man erkennt deutlich den Wechsel von mehreren metastabilen Phasen. Nach etwa 8000 Zeitschritten kommen die Gene H und O auf, die schon als stabile Phase aus Abbildung 5.9 bekannt sind.

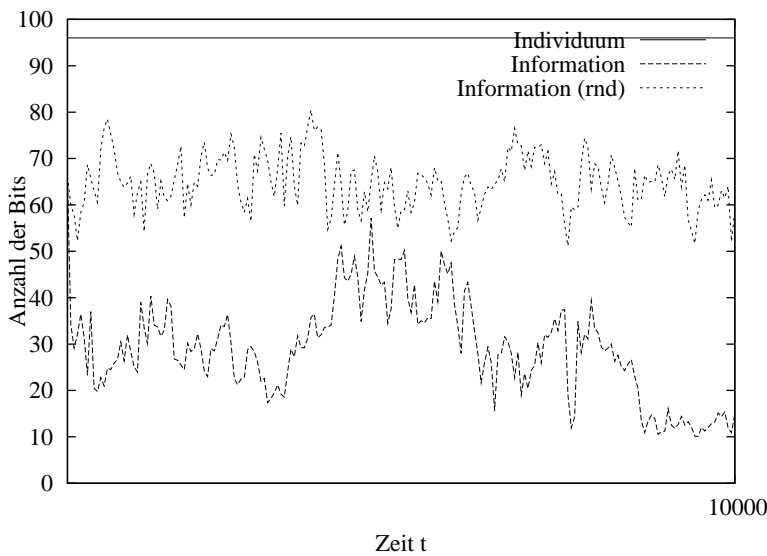
5.5.4 Die Informationslänge

Während zu Beginn der Simulation jeder Bit-String etwa 11 Gene enthält, sind es am Ende der Simulation nur noch etwa 1 bis 2 Gene. Damit besteht ein Individuum zu etwa 90% aus Junk. Man sieht dies schon in Abbildung 5.12. Noch deutlicher wird es allerdings, wenn man die zeitliche Entwicklung der Informationslänge betrachtet, die in Abbildung 5.13 dargestellt wird. Innerhalb von 50 Zeitschritten sinkt die Informationslänge von 66 Bits auf etwa 30 Bits. Bis etwa zum Zeitpunkt $t = 4000$ bleibt die Informationslänge unter 40 Bits, die nächsten 2000 Zeitschritte pendelt sie zwischen 40 und 50 Bits, bevor sie wieder auf unter 40 Bits absinkt. Sobald etwa zum Zeitpunkt $t = 8000$ die sehr stabile Phase aus den Genen H und O auftritt, sinkt die Informationslänge sogar auf unter 15 Bits ab. Im Gegensatz dazu schwankt die Informationslänge bei einem Zufallsmodell um den Mittelwert von etwa 66 Bits (gestrichelte Linie in Abbildung 5.13).

Diese Tendenz zu geringem Informationsgehalt ist ganz typisch bei der Verwendung von Startcodons. Dies zeigt Abbildung 5.14, in der die relative Informationslänge, das ist die Informationslänge normiert auf die Länge der Individuen, dargestellt wird. Während bei einem Zufallsmodell die normierte Informationslänge etwa bei $2/3$ liegt, beträgt sie hier weniger als $1/2$ und geht in Extremfällen sogar auf 15% herunter.

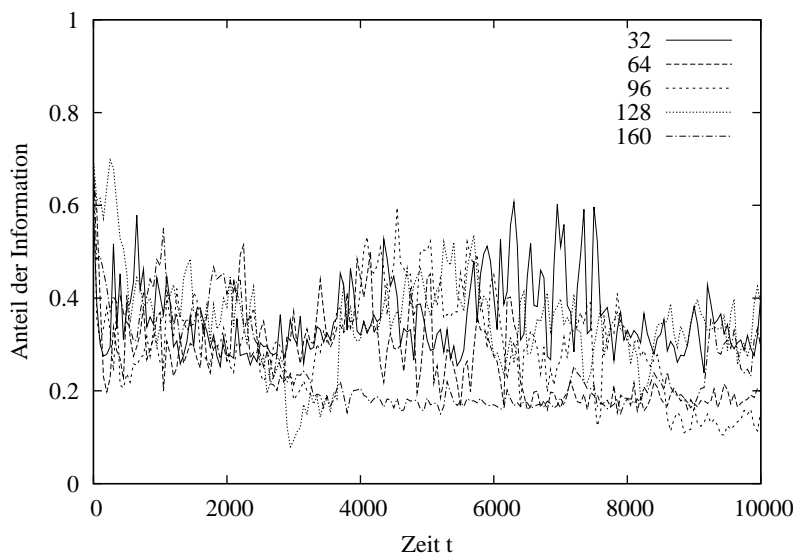
Wie kann man diesen Informationsverlust verstehen? Bei den Simulationen, die zu Abbildung 5.14 geführt haben, ist pro Replikation genau eine Punktmutation durchgeführt worden. Gehört das mutierte Bit zu einem Gen oder Startcodon, dann ist der Nachkomme verschieden von seinem Vorfahren. Liegt dagegen das mutierte Bit in einer Junkregion, dann muß schon ein Startcodon erzeugt werden, damit Vorfahr und Nachkomme zu einer

Abbildung 5.13: Zeitliche Entwicklung der Informationslänge



Diese Abbildung zeigt die zeitliche Entwicklung der mittleren Informationslänge für den Lauf aus Abbildung 5.12. Die durchgezogene Linie zeigt die Gesamtlänge, die konstant bei 96 Bits liegt. Die gestrichelte Linie zeigt die Informationslänge die bei etwa 66 Bits startet und zum Schluß auf etwa 15 Bits absinkt. Die gepunktete Linie zeigt im Vergleich dazu, die Entwicklung der Informationslänge bei einem Zufallsmodell, sie schwankt um den Mittelwert von etwa 66.

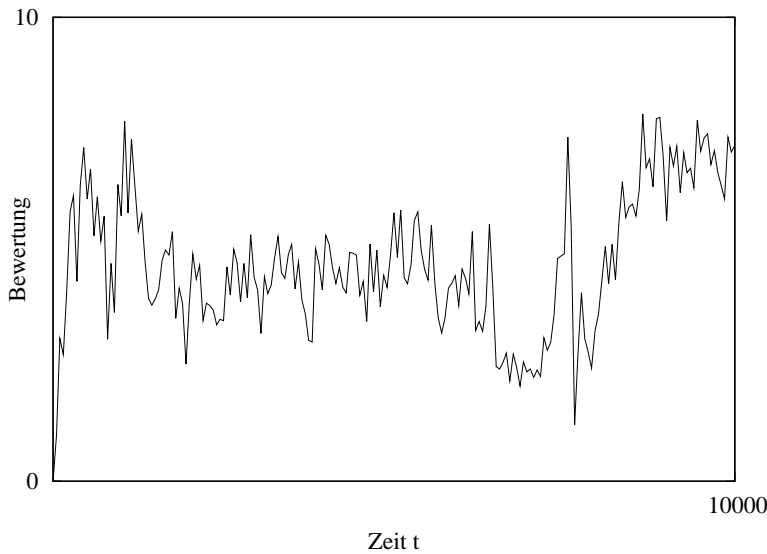
Abbildung 5.14: Informationsgehalt bei verschieden langen Bit-Strings



In dieser Abbildung wird die zeitliche Entwicklung der normierten, mittleren Informationslänge bei Verwendung von Startcodons verglichen. Die Informationslänge wird auf die Gesamtlänge der Bit-Strings normiert. Es werden Bit-Strings der Länge 32, 64, 96, 128 und 160 Bits verglichen. Bei einem Zufallsmodell beträgt die durchschnittliche Informationslänge etwa $2/3$.

unterschiedlichen Dekodierung (d.h. zu unterschiedlichen Spezies) führen. Enthält also ein Bit-String sehr wenig Information, so gehört nach einer Punktmutation der Nachkomme mit großer Wahrscheinlichkeit zur selben Spezies wie sein Vorfahr. Da nur *gute* Bit-Strings replizieren und da Mutationen, die den Geninhalt verändern, normalerweise zu schlechteren Individuen führen, ist es von Vorteil, identische Nachkommen zu erzeugen.

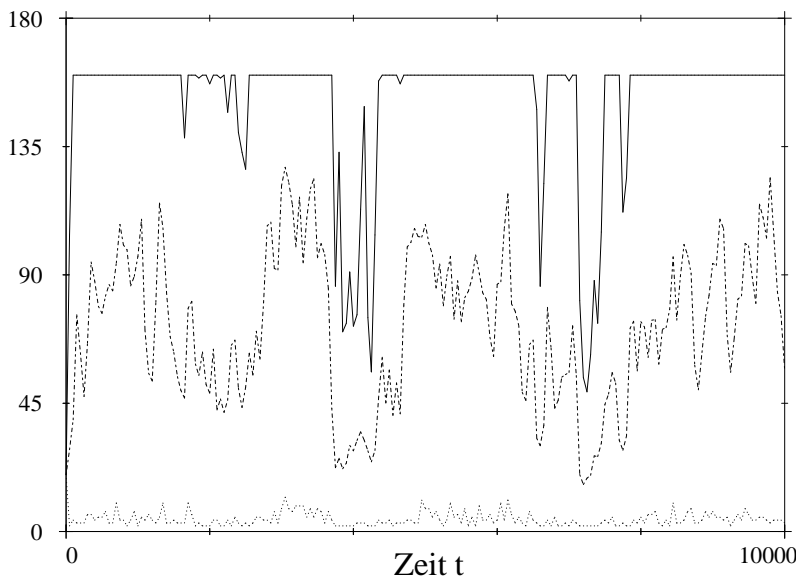
Kann es nun geschehen, daß Individuen entstehen und sich durchsetzen, die nur Junk und keine Information enthalten? Da die Einträge der Wechselwirkungsmatrix im Intervall $[-1, 1]$ liegen, liegt diese Vermutung nahe. Betrachtet man aber die mittlere Bewertung der Bevölkerung, die in Abbildung 5.15 dargestellt wird, so erkennt man sofort, daß Individuen ohne Gene keine Überlebenschance haben. Sie haben als Bewertung 0.0, daher sind sie wesentlich schlechter als die durchschnittliche Bevölkerung und werden eliminiert.

Abbildung 5.15: Zeitliche Entwicklung der Bewertung

In dieser Abbildung wird die zeitliche Entwicklung der Bewertung der Bevölkerung dargestellt. Die Daten stammen aus derselben Simulation, die den Abbildungen 5.12 und 5.13 zugrunde liegt. Die maximale Bewertung kann +10 die minimale -10 betragen. Im Zufallsmodell pendelt sie um den Wert Null herum.

5.5.5 Variation der Individuenlänge

Da die Informationslänge eine deutliche Tendenz zu niedrigen Werten aufweist, kann man sich Fragen, was passiert, wenn die Länge der Bit-Strings selbst verändert werden kann. Bisher wurde als Quelle der Veränderung nur eine einzelne Punktmutation

Abbildung 5.16: Entwicklung der Bitlänge bei Verwendung von Cut-and-Splice

Die drei Kurven zeigen (von oben nach unten) die maximale Bitlänge (die höchstens 160 betragen konnte), die mittlere Bitlänge und die minimale Bitlänge (die mindestens 2 betragen konnte) in einer Population von 64 Individuen. Die mittlere Bitlänge zeigt starke Fluktuationen um den mittleren Wert von 80 Bits. (Entnommen aus [42])

zugelassen. Zusätzlich kann aber auch noch der in Abschnitt 3.4.3 beschriebene Cut-and-Splice-Operator verwendet werden, der Längenänderungen der Bit-Strings zuläßt. Um den Einfluß dieses Operators zu testen, wurden Simulationen durchgeführt, bei denen eine minimal Länge von 2 und eine maximale Länge von 160 Bits erlaubt war. Abbildung 5.16 zeigt die zeitliche Entwicklung der minimalen (gepunktete Linie), der mittleren (gestrichelte Linie) und der maximalen Bitlänge (durchgezogene Linie) in einer Population von 64 Bit-Strings. Als Startlänge wurde willkürlich 20 Bits gewählt. In jedem Zeitschritt wurde die Hälfte der neu erzeugten Bit-Strings mit Cut-and-Splice aus zwei Vorfahren generiert, während die andere Hälfte, wie sonst üblich, aus je einem Vorfahren unter Verwendung des Mutationsoperators erzeugt wurde. Man sieht an dieser Abbildung, daß die

mittlere Bitlänge sehr starke Fluktuationen um den mittleren Wert von 80 Bits herum aufweist. Daher kann man weder einen generellen Trend zu kurzen, noch zu langen Individuen feststellen. Bei der gewählten Implementation existiert also kein selektiver Vorteil für irgendeine Länge, mit anderen Worten ausgedrückt, *es findet neutrale Selektion statt* [27, 56, 57].

Abbildung 5.17: Länge der Anfangspopulation 20 Bits

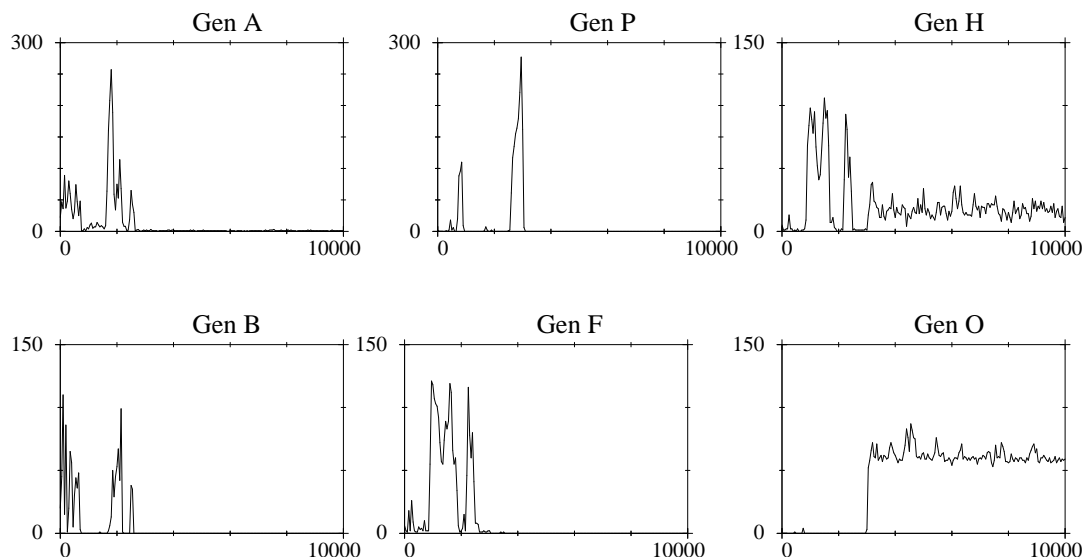


Abbildung 5.17 und 5.18 zeigen die Genbesetzungszahlen für die sechs erfolgreichsten Gene. Es wurden Startcodons und das Sortiermodell mit 64 Individuen verwendet. Cut-and-Splice erlaubt Längenänderungen der Individuen. Die maximal erlaubte Länge beträgt 160 Bits. Diese und die die nächste Abbildung zeigen, wie sensitiv die Ergebnisse von den Anfangsbedingungen abhängt.

Diese Abbildung stammt aus einer Simulation, bei der die Individuen der Anfangspopulation eine Länge von 20 Bits haben. Zu Beginn herrschen große Fluktuationen vor. Man erkennt die kurzzeitige Dominanz der Gengruppen A, B und F, H, wie in Abbildung 5.10. Gen P hat zwei Ausbrüche. Nach dem stärkeren zweiten erscheint ein (meta-)stabiler Zustand, der die Gene H und O enthält. (Entnommen aus [42])

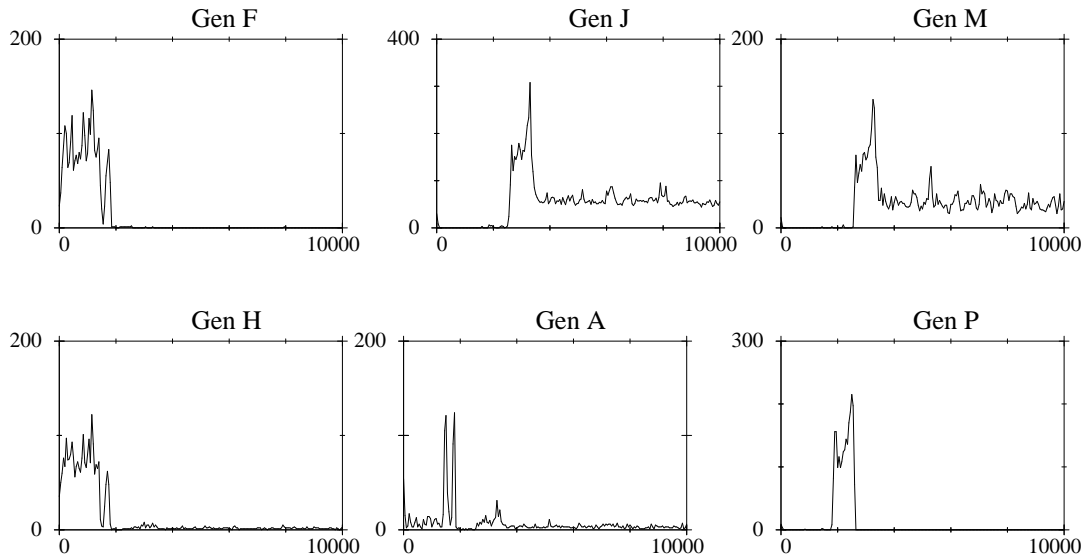
5.5.6 Abhängigkeit von den Anfangsbedingungen

Im vorigen Abschnitt wurde gezeigt, daß die Bitlänge starken Schwankungen unterliegt. Abbildung 5.17 macht deutlich, daß dies auf das qualitative Verhalten keinen Einfluß hat. In dieser Abbildung sieht man die Genbesetzungszahlen für einige der 20 Gene aus der Simulation, die auch schon Abbildung 5.16 zugrunde lag. Obwohl während der Dominanz der Gene H und O die mittlere Bitlänge zwischen 20 und 120 Bits schwankt, zeigen die Besetzungszahlen nur geringe Schwankungen. Man sieht also deutlich, daß die eigentlich interessante Größe die Informationslänge ist, denn sie bleibt konstant niedrig.

Dennoch läßt sich diese Variante des Bit-String-Modells sehr schön dafür verwenden, um zu zeigen, wie sensitiv das ganze System auf die Anfangsbedingungen reagiert. Während in Abbildung 5.17 als Startlänge 20 Bits gewählt wurde, betrug die Startlänge in Abbildung 5.18 dagegen 60 Bits. In beiden Fällen führt eine kurze Dominanz des Gens P zum Aufkommen einer stabilen Phase (Gene H und O in Abbildung 5.17 und Gene J und M in Abbildung 5.18).

Aus den obigen Ausführungen wird klar, daß es nicht nötig ist, Längenvariationen bei den Bit-Strings zuzulassen. Der interessante Effekt, die Tendenz zu kurzen Individuen (gemessen in der Informationslänge), wird auch bei konstanter Individuenlänge deutlich.

Abbildung 5.18: Länge der Anfangspopulation 60 Bits



Diese Abbildung stammt aus Simulationen, die mit den selben Parametern durchgeführt wurden, wie die, die zu Abbildung 5.17 geführt haben. Der einzige Unterschied besteht darin, daß Individuen der Anfangspopulation eine Länge von 60 Bits haben. Die Tendenz ist ähnlich, auch hier markiert die kurzzeitige Dominanz des Gens P einen Übergang in einen metastabilen Zustand, der nun allerdings aus den Genen J und M besteht. (Entnommen aus [42])

Daher wird im Folgenden auf die Verwendung des Cut-and-Splice-Operators verzichtet, so daß die Individuenlänge in allen Simulationen konstant bleibt. Veränderungen werden nur durch den Mutationsoperator und in einzelnen Fällen durch Crossover erzeugt.

5.5.7 Speziesspuren und evolutionäre Aktivität

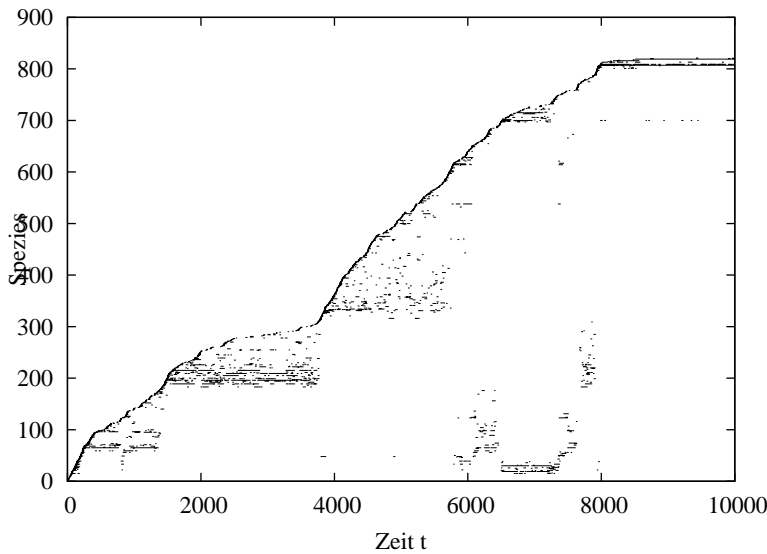
Untersucht man die langlebigen Individuen genauer, so stellt man fest, daß einzelne Sequenzen immer wieder auftauchen. Dies führt zum Konzept der Spezies, wie es in Abschnitt 3.5 definiert wurde. In Abbildung 5.19 werden *Speziesspuren* für alle Spezies dargestellt, die mindestens einmal 20 Zeitschritte in der Simulation vorhanden waren. Die Spezies werden in der Reihenfolge ihres ersten Auftretens in der Simulation nummeriert.

Man erkennt, wie auch schon in Abbildung 5.12, den Wechsel mehrerer metastabiler Phasen. Bei dieser Art der Darstellung wird auch klar, daß man in dieser Simulation die schon besprochenen Phänomene *Genetic takeover* (siehe Abbildung 5.9) und *Bistabilität* (siehe Abbildung 5.10) wiederfindet. In Abbildung 5.19 sieht man, daß mehrere Spezies der ersten länger stabilen Phase (sie haben die Nummern von etwa 70 bis 200) dreimal auftauchen. Auch die Spezies der anderen Phasen können wiederkehren.

In Abbildung 5.19 fällt weiterhin auf, daß die Übergänge ziemlich scharf sind. Während die einzelnen Phasen einige hundert bis etwa 2000 Zeitschritte andauern, findet der Wechsel innerhalb weniger Zeitschritte statt (Stichwort: *Punctuated Equilibria* [31, 85]). Dies erinnert an Fossilien Daten im Zusammenhang mit den Massensterben der Erdgeschichte [1, 103]. Obwohl in diesem Modell keine von außen eingeführten Katastrophen vorkommen, beobachtet man trotzdem so etwas wie *Massensterben* (bei einer Population von 64 Individuen ist dieser Begriff sicher übertrieben). Damit erhält man ähnliche Ergebnisse wie KAUFFMAN und JOHNSEN [71] und LINDGREN [85].

Sieht man sich nun die *einshüllende Kurve* etwas genauer an, so stellt man fest, daß unter-

Abbildung 5.19: Speziesspuren



Diese Abbildung zeigt Speziesspuren für die Simulation, der auch die Abbildungen 5.12, 5.13 und 5.15 zugrunde lagen. Die Spezies werden in der Reihenfolge ihres ersten Auftretens in der Simulation numeriert. Es werden alle Spezies berücksichtigt, die mindestens einmal länger als 20 Zeitschritte vorhanden waren.

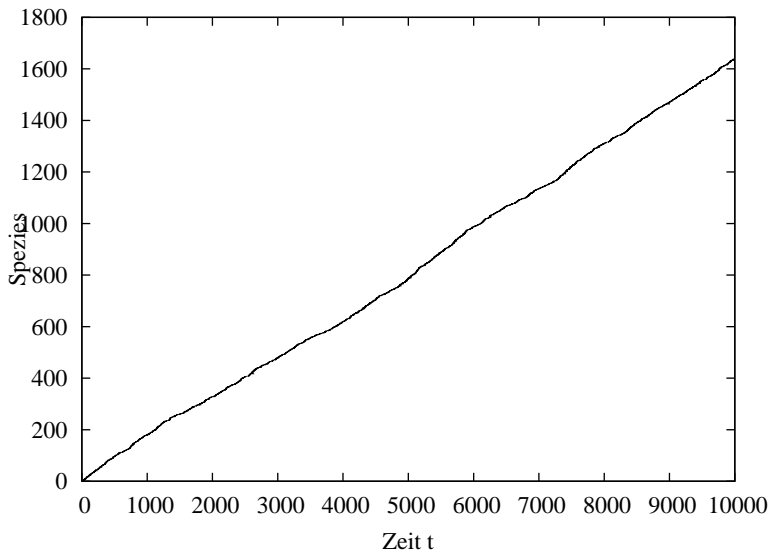
schiedliche Steigungen existieren. Während nach dem Entstehen der letzten metastabilen Phase, deren Spezies aus den Genen H und O besteht (siehe Abbildung 5.12), keine länger lebenden Spezies mehr aufkommen, werden während der dritten metastabilen Phase (sie beginnt bei $t \simeq 4000$) ständig neue Spezies erzeugt. Von den nach $t \simeq 4600$ entstehenden Spezies kann sich aber keine länger in der Simulation halten. Es sieht also danach aus, daß diese neuen Spezies die Phase instabil machen und dafür sorgen, daß eine Serie von Übergängen eingeleitet wird, die schließlich bei $t \simeq 8000$ in der sehr stabilen letzten Phase mündet.

In Abschnitt 6.3 im nächsten Kapitel wird der Begriff *evolutionäre Aktivität* eingeführt. An dieser Stelle soll keine Definition angegeben werden, es genügt, wenn man eine intuitive Vorstellung von dem Begriff hat. Die in Abbildung 5.19 dargestellten Daten lassen es natürlich erscheinen, die Steigung der Einhüllenden zu verwenden, um ein Maß für evolutionäre Aktivität zu definieren. Eine flache Einhüllende (also Steigung Null) deutet an, daß die Evolution zum Stillstand gekommen ist. Eine positive Steigung zeigt, daß noch Evolution im Gange ist. Man kann allerdings auch die Steigung selber nicht als Maß verwenden. Dies wird klar, wenn man Abbildung 5.20 betrachtet. Dort werden die Speziesspuren für ein Zufallsmodell gezeigt. Die Einhüllende kann diesmal sehr gut durch eine Gerade angenähert werden. Die Steigung entspricht der maximalen Steigung der Einhüllenden in Abbildung 5.19. Im Zufallsmodell findet also ständiger Wechsel statt, es kann sich aber keine der Spezies länger halten. Es passiert nicht einmal, daß eine Spezies wieder auftaucht, ganz im Gegensatz zu der nicht zufälligen Simulation aus Abbildung 5.19. Dieses *Rauschen* würde man also genausowenig als Evolution bezeichnen, wie den Fall ohne Entstehung neuer Spezies.

Damit man zu recht sagen kann, ein System zeigt evolutionäre Aktivität, sollten sich stabile Phasen und Phasen der Veränderung abwechseln (in diesem Zusammenhang wird in der Literatur der Begriff der *durchbrochenen Gleichgewichte* verwendet [31, 85]). Dies wird in einer sich ständig ändernden Steigung der Einhüllenden widerspiegelt. Anders ausgedrückt heißt das, sobald die Steigung der Einhüllenden konstant wird (also den maximalen Wert oder den Wert Null annimmt), findet keine Evolution mehr statt. Eine ständig wechselnde Steigung ist also am ehesten ein Indiz für evolutionäre Aktivität.

Welche Schwierigkeiten die Definition des Begriffes *evolutionäre Aktivität* bereitet, wird

Abbildung 5.20: Speziesspuren im Zufallsmodell



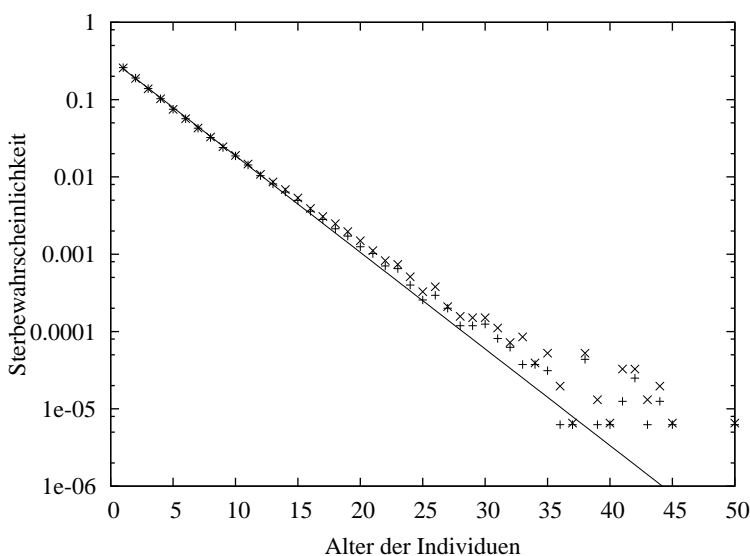
Diese Abbildung zeigt Speziesspuren für ein Zufallsmodell, die dazugehörige Informationslänge ist die gestrichelte Linie in Abbildung 5.13. Die Spezien werden in der Reihenfolge ihres ersten Auftretens in der Simulation numeriert. Es werden alle Spezien berücksichtigt, die mindestens einmal länger als 20 Zeitschritte vorhanden waren. Weitere Ausführungen finden sich im Text.

in Abschnitt 6.3 noch deutlicher, daher wird an dieser Stelle auf eine weitere Diskussion verzichtet.

5.6 Die Bedeutung von Spezien

Bei den Simulationen, die in den Abschnitten 5.3 und 5.4 vorgestellt wurden, ist bei der Erzeugung von Nachwuchs grundsätzlich eine Punktmutation durchgeführt worden. Bei der Verwendung des Begriffes *Spezien*, wie er in Abschnitt 3.5 eingeführt wurde, hat dies zur Folge, daß ein Individuum bei der Replikation einen Bit-String erzeugt, der zu einer anderen Spezien gehört (vergleiche die Abbildungen 5.4 und 5.7). In Abbildung 5.21 wird die Individuenaltersverteilung (Rhomben) mit der Speziessaltersverteilung (Kreuze) ver-

Abbildung 5.21: Vergleich Individuen- und Speziessaltersverteilung

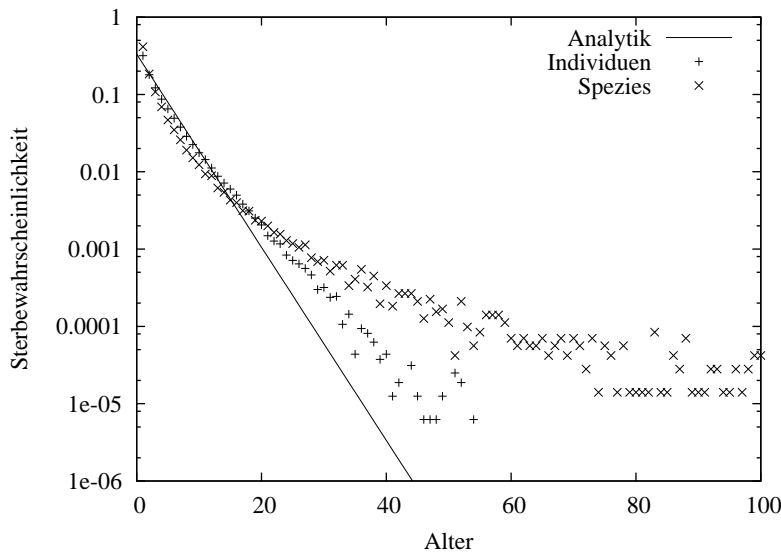


Diese Abbildung zeigt die Altersverteilung für Individuen (Rhomben) und Spezien (Kreuze) bei Verwendung des Dekodierschemas aus Abschnitt 5.3, wenn bei jeder Replikation eine Punktmutation durchgeführt wurde. Es wurden 64 Bit-Strings über eine Laufzeit von $t_{sim} = 10\,000$ Zeitschritten simuliert (dies entspricht den Daten für $\alpha = 10$ aus Abbildung 5.5). Die durchgezogene Linie entspricht Gleichung (A.1) für $p = 0.25$. Man kann Individuen und Spezien nicht unterscheiden.

glichen. Die durchgezogene Linie entspricht der theoretischen Altersverteilung für ein Zufallsmodell nach Gleichung (A.1). Man erkennt deutlich, daß es keinen Sinn macht, zwischen Individuen und Spezien zu unterscheiden. Erst ein Nachkomme in zweiter Generation (quasi ein *Enkel*) kann zur selben Spezien gehören wie der *Großvater*, aber nur,

wenn die beiden Punktmutationen am selben Bit stattfinden.

Abbildung 5.22: Vergleich Individuen- und Speziesaltersverteilung



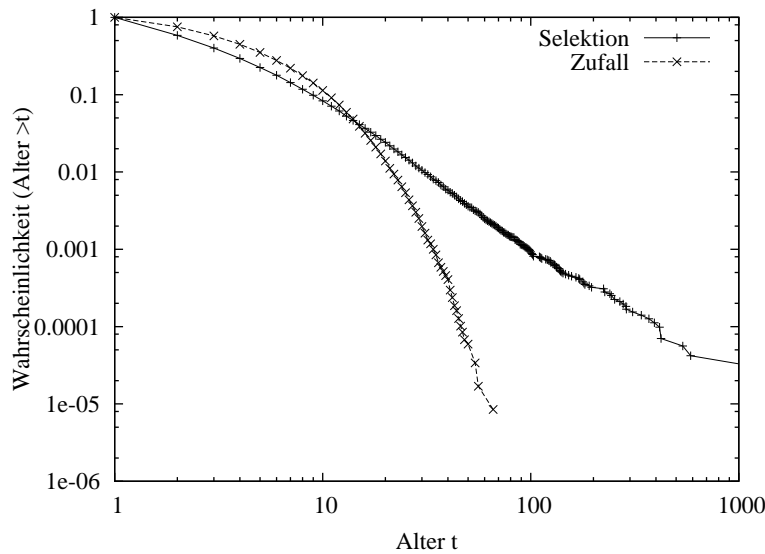
Diese Abbildung zeigt die Altersverteilung für Individuen (Rhomben) und Spezies (Kreuze) bei Verwendung von Startcodons. Die durchgezogene Linie entspricht Gleichung (A.1). Die Daten stammen aus derselben Simulation wie die Abbildungen 5.12, 5.13, 5.15 und 5.19. Es existieren noch 63 Spezies, die länger als 100 Zeitschritte in der Simulation vorkamen. (Weitere Erläuterungen befinden sich im Text.)

Ein ganz anderes Bild liefern die Simulationen, die bei Verwendung von Startcodons durchgeführt wurden. Als Beispiel dienen hier wieder die Daten, die schon zu den Speziesspuren aus Abbildung 5.19 geführt haben. In Abbildung 5.22 wird wieder die Individuenaltersverteilung (Rhomben) mit der Speziesaltersverteilung (Kreuze) verglichen. Diesmal ist ein deutlicher Unterschied sichtbar. Während die Individuenaltersverteilung bei $t \simeq 50$ abbricht, existieren noch über 60 Spezies, die mindestens 100 Zeitschritte in der Simulation vorlagen. Die Spezies liefern nun also neue Informationen, die vorher nicht zugänglich waren.

Die Individuenaltersverteilung ist sehr gut verstanden, sowohl die Exponentialverteilung bei dem Zufallsmodell (siehe Anhang A), als auch die Abweichungen davon bei wirksamer Selektion (siehe Abschnitt 5.3.3).

Kann man nun für die Speziesaltersverteilung ähnliche Rechnungen durchführen? Es bietet sich hierzu an, die Theorie der Verzweigungsprozesse zu verwenden [55]. Die Rechnungen in Anhang B zeigen, daß man dann asymptotisch ebenfalls eine Exponentialverteilung erwarten sollte. Simulationen mit Zufallsmodellen bestätigen dies auch (siehe Abbildung B.1). Dort gilt allerdings auch in erster Näherung die Voraussetzung, daß die Individuen nicht miteinander in Wechselwirkung treten. Bei biologischen Populationen und speziell auch beim Bit-String-Modell ist es aber gerade die Wechselwirkung, die untersucht werden soll. Daher ist zu vermuten, daß die Theorie der Verzweigungsprozesse nur von bedingter Relevanz sein wird [55]. Dies wird bestätigt, wenn man sich Abbildung 5.23 ansieht. Dort wird die Speziesaltersverteilung aus Abbildung 5.22 mit den Ergebnissen aus einem Zufallsmodell verglichen. Bei einer halblogarithmischen Darstellung befinden sich die Werte des Zufallsmodells auf einer Geraden. Es liegt also eine Exponentialverteilung vor. In Abbildung 5.23 ist dagegen eine doppeltlogarithmische Darstellung gewählt worden. In dieser Darstellungsweise fällt eine Exponentialfunktion sehr schnell ab, wie man an den Kreuzen des Zufallsmodells erkennen kann. Die Daten aus der Simulation mit Selektion lassen sich dagegen eher durch ein Potenzgesetz beschreiben, denn sie liegen in etwa auf einer Geraden.

Dieses Potenzgesetz könnte ein erstes Anzeichen für eine beginnende offene Evolution sein. Das Potenzgesetz deutet nämlich an, daß alle Zeitskalen bei der Speziesaltersverteilung

Abbildung 5.23: Speziesaltersverteilung im Vergleich mit dem Zufallsmodell

Diese Abbildung zeigt die Speziesaltersverteilung (Rhomben) im Vergleich zum Zufallsmodell (Kreuze) bei Verwendung von Startcodons. Um die Tendenz zu verdeutlichen, wird auf der y -Achse die Wahrscheinlichkeit angezeigt, daß eine Spezies mindestens t Zeitschritte überlebt hat. (Weitere Informationen befinden sich im Text.)

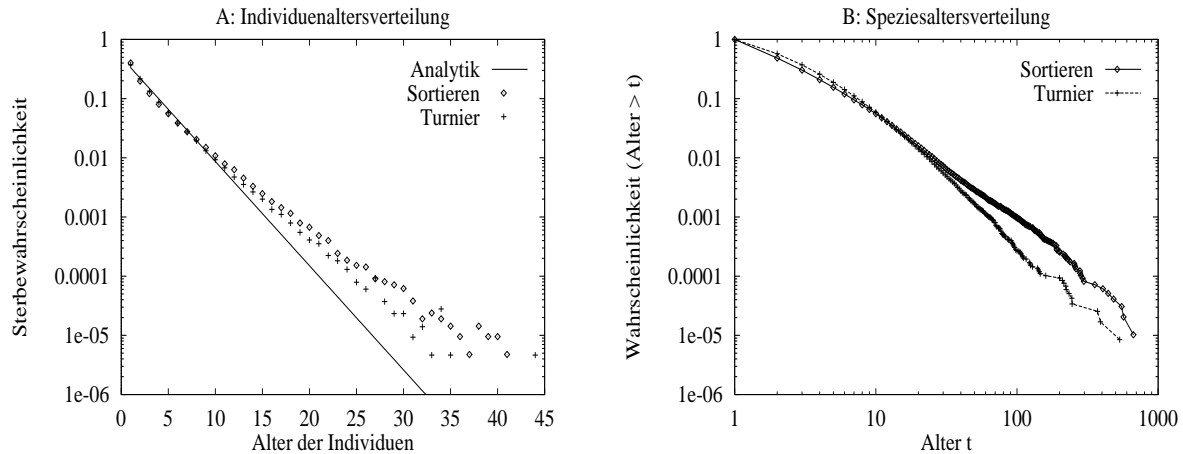
vertreten sind. Damit hat man die schon mehrfach angesprochenen langen stabilen Phasen **und** gleichzeitig auch immer den Wechsel, der es erlaubt, daß neue Spezies aufkommen und dominieren.

5.7 Variationen des Modells

Die in diesem Kapitel diskutierten Simulationen wurden alle mit dem Sortiermodell (siehe Abschnitt 3.4.2) durchgeführt. Bei der Erzeugung von Nachwuchs wurde immer eine Punktmutation durchgeführt. Speziell der zweite Punkt ist eine starke Einschränkung. Es ist viel sinnvoller, eine bestimmte Mutationsrate p_{mut} pro Bit zu definieren (siehe Abschnitt 3.4.3 und speziell Gleichung (3.6)). Eine Mutationsrate von $p_{mut} = 10^{-3}$ bewirkt bei einer Individuenlänge von 96 Bits, daß bei etwa 91% aller Replikationsvorgänge identischer Nachwuchs erzeugt wird. Verwendet man diesen Mutationsoperator, so haben schon beim Dekodierschema *Gen an Gen* Individuen und Spezies deutlich verschiedene Altersverteilungen.

Bei der Definition des Bit-String-Modells wurde großer Wert darauf gelegt, daß globale Eigenschaften der Population keinen unmittelbaren Einfluß auf einzelne Individuen haben (siehe die Kriterien von LANGTON in Kapitel 2). Das Sortiermodell ist aber ein **globales** Selektionsschema, da die Bewertungen aller Individuen verglichen werden. Als **lokales** Selektionsschema – der Begriff *lokal* wird hier nur als Gegenpol zu *global* und nicht als räumliche Charakterisierung verstanden – bietet sich die Verwendung des Turniermodells an (siehe Abschnitt 3.4.2). Dies hat natürlich Rückwirkungen auf die Individuenaltersverteilung: Während im Sortiermodell sehr starke Abweichungen von der theoretischen Verteilung nach Gleichung (A.1) vorliegen, liegen bei dem lokaleren Schema geringfügigere Abweichungen vor (siehe Abbildung 5.24A). Im Unterschied zu den bisherigen Abbildungen wurde auch für das Sortiermodell als Eliminationswahrscheinlichkeit $p = 1/3$ gewählt, damit die Daten vergleichbar mit denen des Turniermodells sind (siehe die Berechnung der Eliminationswahrscheinlichkeit in Abschnitt 3.4.2 und speziell Gleichung (3.5)). Die Speziesaltersverteilungen zeigen ebenfalls geringfügige Unterschiede zwischen Sortier- und Turniermodell. Die Tendenz bleibt aber dieselbe; man kann die Verteilung eher durch ein Potenzgesetz als durch ein Exponentialgesetz beschreiben (siehe Abbildung

Abbildung 5.24: Vergleich Sortiermodell und Turniermodell



Diese Abbildung zeigt im Vergleich die Individuenaltersverteilung (links) und Speziesaltersverteilung (rechts) für das Sortiermodell (Rhomben) und das Turniermodell (Kreuze). Die Populationsgröße beträgt in beiden Fällen 64 Individuen von je 96 Bits, die Simulationsdauer $t_{sim} = 10000$ Zeitschritte. Die Eliminationswahrscheinlichkeit ist in beiden Fällen $p = 1/3$.

5.24B). Da weder der andere Selektionsmechanismus noch der veränderte Mutationsoperator zu qualitativ neuen Ergebnissen führen, wird auf eine ausführlichere Besprechung von Simulationsergebnissen verzichtet.

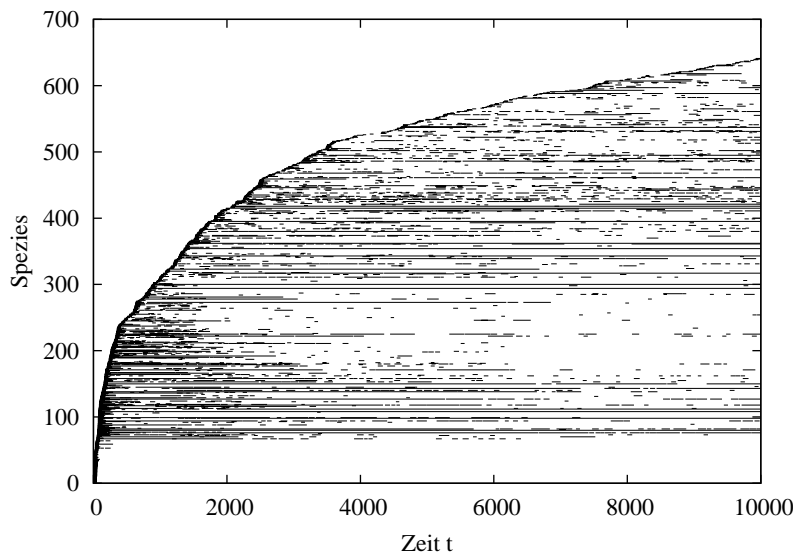
Bisher bestand die Bevölkerung aus 64 Individuen. Dies ist eine sehr geringe Zahl. Eine relativ kleine Fluktuation kann daher schon zum Aussterben einer Spezies führen. Daher ist es interessant zu untersuchen, wie sich eine sehr große Population ($N \simeq 10^4$) verhält, ob die dort größere Zahl der Individuen eine größere Stabilität von Spezies zur Folge hat. In Abbildung 5.25 sind die Speziesspuren für eine Population von 10 000 Bit-Strings dargestellt. Es sind alle Spezies aufgeführt, die mindestens 50 Zeitschritte in der Simulation vorlagen. Im Unterschied zu den bisher beschriebenen Simulationen findet kein Wechsel mehrerer metastabiler Phasen statt. Die hohe Bevölkerungsgröße erlaubt es, daß zu Beginn mehrere Spezies sehr stark werden. Während einer typischen Simulation bestehen von den etwa 200 verschiedenen Spezies 10 bis 15 aus jeweils mindestens 100 Mitgliedern. Die stärkste Spezies hat sogar etwa 3 000 Mitglieder.

Diese hohen Besetzungszahlen machen also, wie erwartet, die einzelnen dominanten Spezies sehr stabil. Dies gilt besonders dann, wenn das Sortiermodell verwendet wird. Eine neue Spezies hat große Schwierigkeiten so stark zu werden, daß eine Fluktuation sie nicht mehr zerstören kann. Ein ähnlicher Effekt ergibt sich, wenn die Wechselwirkungsrate α sehr groß wird. Dies unterdrückt ebenfalls Fluktuationen. Innerhalb von 500 bis 2000 Zeitschritten bildet sich eine sehr stabile Phase aus. Diese stabile Phase kann durch eine Quasispezies charakterisiert werden.

Umgekehrt ausgedrückt ergibt es sich, daß das Bit-String-Modell besonders dann interessante Ergebnisse liefert, wenn die Fluktuationen ständige Wechsel zwischen verschiedenen Quasispezies erlauben. Dies ist bei kleiner Bevölkerungsgröße und moderater Wechselwirkungsrate der Fall.

Obwohl ein potentiell sehr großer Phasenraum zur Verfügung steht, läßt der gewählte Wechselwirkungsmechanismus nicht zu, daß dieses Potential vollständig ausgeschöpft wird. Bei allen Simulationen wird immer nur eine relativ kleine Anzahl von Spezies gefunden, die sich als erfolgreich erweisen. Diese können immer als die Mitglieder einer

Abbildung 5.25: Speziesspuren für eine sehr große Population



Diese Abbildung zeigt Speziesspuren für eine Simulation mit einer Bevölkerung von 10 000 Bit-Strings. Es wurde das Turniermodell verwendet. Die Wechselwirkungsrate α betrug 10.

Quasispezies aufgefaßt werden.

Das Konzept der gut durchmischten Flüssigkeit, das in dieser Arbeit verwendet wurde, verhindert Nischenbildung. Dies ist vermutlich eine Ursache dafür, daß sich immer nur eine Quasispezies ausgebildet hat. Es ist zwar möglich, daß die Simulation eine Folge von Quasispezies ergibt, aber nur in der kurzen Phase des Übergangs sind zwei Quasispezies vorhanden. Eine mögliche Erweiterung des Modells bestünde daher darin, die Bit-Strings auf ein Gitter zu setzen. Diese zusätzliche räumliche Umgebung würde Nischenbildung ermöglichen. Jeder Gitterplatz könnte dabei von einem oder mehreren Individuen besetzt werden. Wechselwirkungen würden dabei nur zwischen räumlich benachbarten Bit-Strings erlaubt werden. In jedem Zeitschritt könnte zusätzlich eine begrenzte Bewegung der Bit-Strings erfolgen. Dies könnte vermutlich sehr effektiv auf einem Parallelcomputer wie der Connection Machine implementiert werden. Als die Arbeiten an dem Bit-String-Modell begonnen wurden, stand eine solche Maschine noch nicht zur Verfügung. Mittlerweile wäre die Möglichkeit zur Benutzung einer CM2 an diesem Institut vorhanden. Allerdings hätte die Implementierung des Modells eine umfangreiche Programmänderung erfordert, so daß im Rahmen dieser Arbeit leider darauf verzichtet werden mußte.

5.8 Zusammenfassung

Das in dieser Arbeit untersuchte Bit-String-Modell stellt ein Klasse von sehr einfachen Systemen dar, die den von LANGTON [83] aufgestellten Kriterien für Artificial Life Systemen genügen. Es konnten erste ermutigende Resultate erzielt werden. Leider deuten Langzeitsimulationen über $t_{sim} \geq 10^5$ Zeitschritte an, daß sich in sehr vielen Fällen eine sehr stabile Phase ausbildet. Es wird zwar auch Bistabilität gefunden, aber nur sehr schwache Anzeichen von Multistabilität. Daraus kann geschlossen werden, daß das Bit-String-Modell noch kein minimales Modell darstellt, das offene Evolution ermöglicht.

Es kann sein, daß die Implementation der Wechselwirkung über die Wechselwirkungsmatrix eine zu *lineare* Form der Wechselwirkung darstellt. Es hat sich nämlich herausgestellt, daß man bei den stabilen Phasen die Proportionen der einzelnen enthaltenen Spezies mit Hilfe der Wechselwirkungsmatrix ausrechnen kann. Andere Modelle, wie z.B. das *NK-Modell* von KAUFFMAN [70, 71] haben eine stärker nicht-lineare Wechselwirkung. Leider

gibt es keine *natürliche* Erweiterung des Bit-String-Modells, die auf ein besseres System, das mehr Züge einer offenen Evolution enthält, hoffen läßt.

Eine der vielen möglichen Erweiterungen ist die Einführung eines expliziten Raumes, was weiter oben schon kurz angesprochen wurde. Dieser wird zu einer Reihe von neuen Phänomenen führen. Um entscheiden zu können, welche davon erst durch das Zusammenspiel der wechselwirkenden Population mit der räumlichen Umgebung entstehen, ist es wichtig ein *pures räumliches* System zu untersuchen. Dies wird im nächsten Kapitel gemacht, in dem das Evolving-Bugs-Modell [11, 96] ausführlich diskutiert wird. Dieses ist ein System, in dem keine explizite Wechselwirkung der Individuen stattfindet. Dabei wird es sich als sehr hilfreich erweisen, Zufallsmodelle als Referenzsysteme heranzuziehen, die ja auch in diesem Kapitel ein sehr wichtiges Handwerkszeug waren.

Die Ausführungen in diesem Kapitel haben klar gemacht, daß Rauschen ein sehr wichtiger Bestandteil des Gesamtsystems ist. In diesem Modell wurde das Rauschen von außen in das System gebracht, es war dabei ohne Bedeutung, wie es entsteht. Es stellt sich aber doch die Frage, ob es auch Systeme geben kann, die aus sich selbst heraus Unordnung und Rauschen erzeugen können. In Anhang C wird ein solches System vorgestellt. Es ist vollständig deterministisch und hat doch die charakteristischen Eigenschaften eines Irrweges [41].

6 Das Evolving-Bugs-Modell

Das in dieser Arbeit vorgestellte Bit-String-Modell, das ohne eine explizite, räumliche Umgebung definiert ist, wurde konzipiert als der Versuch, ein minimales Modell zu erschaffen, das evolutionäres Verhalten zeigt. Während in dieser Arbeit der Schwerpunkt auf die Wechselwirkung der Individuen untereinander gelegt wurde, gibt es andere Modelle, die mit dem gleichen Anspruch konzipiert wurden, aber andere Schwerpunkte setzen. Als ein Beispiel für solche Modelle sei hier ausführlich das *Evolving-Bugs-Modell* von PACKARD [4, 11, 96] vorgestellt. Zunächst werden die Eigenschaften angegeben, die allen Varianten des Modells zugrunde liegen, bevor die drei verschiedenen Versionen genauer erläutert werden.

Das Modell-Universum besteht aus einer Population von beweglichen Individuen, die im Folgenden in Anlehnung an die Originalartikel *Bugs* genannt werden. Diese leben auf einem zweidimensionalen, quadratischen Gitter mit periodischen Randbedingungen, so daß eine Torusgeometrie entsteht. Auf dem Gitter befindet sich Futter, das von den Bugs aufgenommen wird und ihnen dann als Energie zur Bewegung und zum Überleben zur Verfügung steht. Ein Bug, dessen Energievorrat zur Neige geht, stirbt und wird aus der Simulation entfernt. Übersteigt der Energievorrat eines Bugs eine bestimmte Schwelle, so erzeugt der Bug Nachwuchs. Insoweit ähnelt das Evolving-Bugs-Modell dem Energieflußmodell von WOLTER [119]. Das Verhalten eines Bugs wird durch *fest vorgegebene* und durch *variable* Regeln definiert. Die variablen Regeln werden als Gene jedes einzelnen Bugs implementiert. Sie sind für einen einzelnen Bug fest vorgegeben, so daß Lernen, wie im Modell von HINTON und NOWLAN [12, 59], *nicht* auf der Ebene von Individuen, sondern *nur* auf der Ebene von Populationen möglich ist. Bugs, die sehr erfolgreich bei der Futtersuche sind, leben lange und erzeugen viele Nachkommen. Andere, die kaum Futter finden, sterben schnell und erzeugen kaum (oder sogar keinen) Nachwuchs.

Damit handelt es sich hier im weitesten Sinne um ein Optimierungsproblem, da der, der am besten in der Lage ist, Futter zu sammeln, die meisten Nachkommen zeugt. Es existiert aber keine extern vorgegebene Fitnessfunktion, deren Optimum zu finden ist, sondern es findet eine *intrinsische Adaption* an die sich (vielleicht) ändernde Umwelt statt, so daß deutliche Ähnlichkeiten zur biologischen Evolution vorhanden sind.

6.1 Einfache Version

Diese erste Version des Evolving-Bugs-Modells zeichnet sich dadurch aus, daß

1. das Futter in regelmäßigen Zeitabständen (gemessen in CPU-Zeit) auf das Gitter regnet,
2. die Bugs keine direkte Wechselwirkung untereinander haben, und
3. jeder Bug nur zwei Gene besitzt.

Die drei Punkte werden im Folgenden näher erläutert:

zu 1: Alle 3 CPU-Sekunden oder alle 0.3 CPU-Sekunden (nur diese beiden Versionen werden in [96] beschrieben) regnet Futter auf das Gitter. Dabei wird ein quadratisches Gebiet von $1/16$ der Gesamtgröße des Gitters *vollständig* gefüllt, d.h., jedes Feld wird auf 255 Einheiten Futter aufgefüllt. Danach findet dreißig Mal pro CPU-Sekunde eine Nächste-Nachbar-Diffusion statt. Da die Futterdynamik, die relativ zu CPU-Sekunden stattfindet, von der Dynamik der Bugs, die in Zeitschritten gemessen wird, entkoppelt ist, kann man für diese Version leider nicht genau feststellen, in welchem Verhältnis ein

Zeitschritt der Bugs und die Zeitkonstante der Futterausgabe stehen. Diese Größe dürfte bei einer schwankenden Bevölkerungsgröße selber starken Schwankungen unterworfen sein. Bei einer kleinen Population benötigt ein Zeitschritt wenig CPU-Zeit, so daß viele Zeitschritte zwischen den Futterausgaben verstreichen. Bei einer großen Population benötigt ein Zeitschritt viel CPU-Zeit, so daß womöglich während eines Zeitschritts mehrfach Futterausgabe stattfindet.

zu 2: Die Bugs wechselwirken nicht miteinander, d.h., jeder Bug kann nur das Futter, aber keine anderen Bugs, in seiner direkten Umgebung feststellen. Der Bug bewegt sich entlang des Futtergradienten *aufwärts*, d.h. in das benachbarte Feld, das das meiste Futter enthält. (Er verwendet also die *gradient ascent* Strategie.) Der Bug frißt ein Feld immer leer. Damit findet natürlich *indirekt* über das Futterfeld auch eine Wechselwirkung der Bugs statt.

zu 3: Die Bewegung und das Fressen jedes Bugs sind von Anfang an fest vorgegeben und ändern sich nicht im Laufe der Zeit. In dieser Version des Evolving-Bugs-Modells gibt es nur zwei variable Größen, die als Gene implementiert sind, zum einen die Schwelle g_{th} , die der interne Futtermvorrat übersteigen muß, damit ein Bug Nachwuchs zeugt, und zum anderen die Anzahl der Nachkommen g_{off} , in die sich ein Bug beim Überschreiten der Schwelle aufteilt. Die Schwelle beträgt zu Beginn 10^4 Einheiten und kann sich bei einer Mutation um 10% ändern, die Zahl der Nachkommen (der erste Nachkomme ist immer eine identische Kopie, alle weiteren Nachkommen werden durch Mutationen verändert) beträgt zu Beginn 2 und kann um 1 zu- oder abnehmen. Neben diesen zur Mutation freigegebenen Größen existiert noch die *Bewegungssteuer*, die bei WOLTER [119] *Energiekosten* genannt wird, sie beträgt 20 bzw. 50 Einheiten. In [96] werden nur diese zwei Varianten diskutiert.

Zeitskalen: Mit den so definierten Größen kann man nun einige theoretische Überlegungen über die Dynamik der Bugs anstellen. Wenn sich ein Bug teilt, wird die Energie gleichmäßig auf alle Nachkommen verteilt, d.h., bei 2 Nachkommen startet jeder Bug mit 5 000 Einheiten Energie. Dies bedeutet, daß er für mindestens 250 bzw. 100 Zeitschritte (je nach Energiekosten) Energie besitzt. Die zweite Zeitskala, die man abschätzen kann, ist die Generationenfolge, d.h. die Zeit, die ein Bug mindestens existieren muß, bevor er selber Nachwuchs zeugt. Da ein Bug mit 5 000 Einheiten startet, benötigt er weitere 5 000 Einheiten, bevor sein Energievorrat die Schwelle g_{th} überschreitet. Da er pro Zeitschritt maximal 255 Einheiten Futter aufnehmen kann, benötigt er je nach Schwelle mindestens 22 bis 25 Zeitschritte. Betrachtet man die Generationenfolge als die relevante Zeitskala, so entsprechen die in [96] dargestellten Simulationen über ca. 20 000 Zeitschritte einer Simulationszeit im Bit-String-Modell von etwa 1 000 Zeitschritten, was ein sehr kurzer Zeitraum ist, der nicht ausreicht, um über die Konvergenz der Dichteverteilung der Population irgendeine definitive Aussage machen zu können.

Meßgrößen: In der ersten Version des Evolving-Bugs-Modells existiert nur eine Meßgröße, die zeitabhängige Dichteverteilung $P^t(g_{th}, g_{off})$ der Bevölkerung über dem zweidimensionalen Raum der Gene. Zu Beginn ist diese Verteilung eine Deltafunktion

$$P^{t=0}(x, y) = \delta(x - 10^4)\delta(y - 2), \quad (6.1)$$

die sich im Laufe der Zeit durch den Einfluß der Mutationen verbreitert. Während der relativ kurzen Simulationsdauer konvergiert $P^t(g_{th}, g_{off})$ nicht gegen eine konstante Verteilung. Ob dies überhaupt möglich ist, konnte in [96] nicht geklärt werden.

Ergebnisse: Die Untersuchung kommt zu zwei wesentlichen Ergebnissen [96]:

1. Wird das Futter in großen Abständen auf das Gitter gebracht (alle 3 CPU-Sekunden), so führt das zu großen Schwankungen in der Bevölkerungsgröße. Dies läßt sich dadurch erklären, daß das Futterfeld selbst großen Schwankungen unterliegt, und daher diese über die Kopplung der Bugs an das Futterfeld (die treibende Kraft) auch auf die Bugspopulation übertragen werden. Außerdem kann es passieren, daß Subpopulationen entstehen, die eine geringe Schwelle besitzen und viele Nachkommen zeugen, so daß eine sehr schnelle Generationenfolge entsteht. Eine solche Subpopulation blüht auf, kurz nachdem Futter auf das Gitter gebracht wird, kann dann aber verhungern, kurz bevor neues Futter kommt. Dies wurde auch beobachtet.
2. Bei einer kürzeren Fütterungsperiode (alle 0.3 CPU-Sekunden) wurde der Einfluß der Energiekosten auf die Dynamik untersucht. Die schnelle Wiederholungsfrequenz der Futterdynamik hat zu geringeren Schwankungen der Bevölkerungsgröße geführt. Hohe Energiekosten (50 Einheiten) hatten ein Aussterben der Bevölkerung nach ca. 12 000 Zeitschritten zur Folge, während bei niedrigen Kosten (20 Einheiten) die Bevölkerung auch nach 20 000 Zeitschritten noch existierte.

Kritik: Die beiden Größen, die hier als Gene implementiert wurden, haben keinen sehr deutlichen Einfluß auf die Überlebensfähigkeit eines Bugs. Das Weiterleben hängt vor allem von den Energiekosten und der Zeitkonstante der Futterdynamik ab, wie man aus den oben dargestellten Ergebnissen ersehen kann. Diese Größen sind aber von außen fest vorgegeben und unterliegen nicht der Evolution.

Eine interessante Eigenschaft, die leider nicht weiter untersucht worden ist, besteht darin, daß Individuen, die genetisch identisch sind, wegen des Vorhandenseins einer räumlichen Umgebung, eine unterschiedliche lokale Fitness haben können.

Die bisher vorgestellte Version stellt ein sehr simples Modell dar, das diverse Variationen erlaubt, von denen zwei in den nächsten beiden Abschnitten besprochen werden.

6.2 Kolonien bildende Bugs

In der ersten Variation, die von ASSAD und PACKARD [4] untersucht wurde, wird eine Wechselwirkung der Bugs definiert. Da sich die Individuen von den oben beschriebenen Bugs stark unterscheiden, werden sie im folgenden in Anlehnung an den Originalartikel [4] *Artorgs* (*artificial organisms* zu deutsch *künstliche Organismen*) genannt.

Die Beschreibung der Artorgs: Im Unterschied zur vorigen Version gibt es mehrere Arten von Artorgs, die durch die Farben *blau*, *grün*, *rot* und *weiß* gekennzeichnet sind. Ein roter Artorg frißt grünes Futter und erzeugt selber rotes Futter, das von einem weißen Artorg gefressen wird, der weißes Futter erzeugt, usw. Die vier Arten bilden eine zirkulare Futterkette. Solange alle Arten vorkommen, wird ständig neues Futter jeder Art erzeugt. Außerdem wird hier, im Gegensatz zur vorigen Version, *nicht* regelmäßig Futter auf das Gitter gegeben. Die Anfangspopulation der Artorgs wird mit einem hohen, internen Energieniveau versehen, so daß jeder Artorg gleich Futter der eigenen Farbe auf das Gitter bringt.

Die Artorgs sind mit mächtigen Sensoren versehen, denn sie können das ihnen am nächsten liegende Futter, auch wenn es sehr weit entfernt liegt, lokalisieren und sich darauf zu bewegen. Ein Artorg kann jede Art von Futter bei sich tragen, aber nur eine Art in Energie umwandeln. Die übrigen drei Arten können aber mit anderen Artorgs, die sich in der Nähe befinden, gehandelt werden. Diese Regelsammlung ist fest vorgegeben. Die einzige variable Handlung besteht darin, welche Art von Futter gegen welche andere Art

gehandelt wird. Dies ist in einer 16 Bit großen Tabelle, die als das *Gehirn* des Artorgs bezeichnet wird, abgespeichert (siehe Tabelle 6.1).

Tabelle 6.1: Beispiel für ein Artorg-Gehirn

	weiß	rot	grün	blau
weiß	1 (e)	0 (k)	0 (n)	0 (n)
rot	1 (u)	0 (u)	1 (u)	1 (u)
grün	0 (n)	0 (k)	0 (e)	1 (n)
blau	1 (n)	1 (k)	0 (n)	0 (e)

Zeilen in der nebenstehende Tabelle geben das Futter an, das ein Artorg mit sich führt, und Spalten das Futter, das ein möglicher Partner mit sich führt. Bits, die auf '1' gesetzt sind, geben an, daß ein Tausch der Futtersorten angestrebt wird. Während Bits, die auf '0' gesetzt sind, anzeigen, daß ein Tausch abgelehnt wird. Die Buchstaben in Klammern werden im Text erläutert. (Entnommen aus [4].)

Die Entwicklung der Handelstabelle: Betrachten wir z.B. einen weißen Artorg und überlegen, welche Strategie, d.h. welche Handelstabelle, für ihn am günstigsten sein wird:

1. Ein weißer Artorg sammelt rotes Futter und verwandelt dieses sofort in Energie. Somit werden die Bits in der *roten* Zeile des Artorg-Gehirns nicht verwendet, sie sind *unbenutzt*, deshalb ist der Eintrag in Tabelle 6.1 ein 'u'. Für diese Bits findet keine Selektion statt, so daß sie im Mittel mit '1' und '0' zu gleichen Anteilen belegt sind, bzw. diesen Zustand im Laufe einer Simulation anstreben.
2. Aus demselben Grund ist es günstig, rotes Futter in jedem Fall zu erhandeln, also sollten diese *kritischen* Bits, die in der roten Spalte mit 'k' gekennzeichnet sind, fast alle mit '1' besetzt sein. Dies findet man nach ca. 3 000 Zeitschritten schon zu 90% bestätigt [4].
3. Die Diagonalelemente der Tabelle, die einen Tausch *Gleiches gegen Gleiches* bezeichnen, können, wie die unbenutzten Bits, ebenfalls nicht der Selektion unterliegen, so daß auch sie im Mittel mit '1' und '0' zu gleichen Anteilen belegt sein werden. D.h., es ist *egal*, welchen Wert sie haben, deshalb das 'e'.
4. Für die übrigen sechs Bits, die mit 'n' für *nicht-kritische* Bits bezeichnet sind, läßt sich keine so eindeutige Aussage machen. Ein Tausch ist nur dann sinnvoll und von Vorteil in der Zukunft, wenn er dazu führt, daß mehr Futter der Sorte erhandelt werden kann, die ein Artorg selber benötigt. Dies hängt aber sehr stark von den Artorgs der näheren Umgebung ab. Hierbei muß man berücksichtigen, daß ein Tausch immer nur dann stattfindet, wenn beide Artorgs zustimmen. Da bei jeder Artorgsorte die Tendenz besteht, daß die kritischen Bits auf '1' stehen, sollten auch die nicht-kritischen Bits mit hoher Wahrscheinlichkeit auf '1' stehen, da nur dann die vorteilhaften Tauschaktionen auch wirklich stattfinden können.

Die Simulationen in [4] deuten an, daß auch bei den nicht-kritischen Bits (auch die durch 'e' gekennzeichneten Bits werden dort als nicht-kritisch bezeichnet) eine Tendenz besteht, daß sich der Anteil der auf '1' gesetzten Bits erhöht. Man erkennt dies daran, daß bei einer zufälligen Initialisierung mit 50% '1' gestartet wird und nach 6 000 Zeitschritten sich dieser Anteil auf 60% erhöht hat. Im gleichen Zeitraum beträgt der Anteil der kritischen Bits, die auf '1' stehen, dann schon mehr als 95% [4]. Man kann also zusammenfassend sagen, daß Handeln immer von Vorteil ist.

Wie man sieht, kann man die Entwicklung einer Handelstabelle relativ leicht vorhersagen. Dies gilt aber nur, wenn man davon ausgeht, daß sich eine oder mehrere Kolonien gebildet haben. Denn nur wenn ein Artorg auf den Nachbarfeldern mögliche Handelspartner hat, wird ein Handel angestrebt.

Die Bildung von Kolonien: Jeder weiße Artorg bewegt sich in Richtung auf das nächste rote Futter. Dies führt dazu, daß sich Paare bzw. kleine Gruppen von weißen Artorgs bilden, die zusammenwachsen. Entsprechendes gilt natürlich auch für rote, blaue und grüne Artorgs. Da rote Artorgs rotes Futter als Abfall hinterlassen, wird sich eine Gruppe weißer Artorgs auf eine rote Kolonie zubewegen, die selbst einer grünen Kolonie hinterherläuft. Dies führt dann in relativ kurzer Zeit dazu, daß sich eine große Kolonie bildet, in der alle Arten von Artorgs vorkommen, obwohl nur die Bewegung auf das nächstliegende Futter bei den Artorgs implementiert ist. In [4] dauert dieser Prozeß 700 Zeitschritte. In dieser Kolonie werden die Handelstabellen der oben beschriebenen Selektion unterworfen sein.

Der Alterungsmechanismus: Die Bugs, die im vorigen Abschnitt vorgestellt wurden, können verhungern, wenn sie nicht genügend Futter finden. Dasselbe gilt natürlich auch prinzipiell für die hier beschriebenen Artorgs. Sobald eine Kolonie gebildet wurde, könnte es nun aber passieren, daß alle zur Kolonie gehörenden Artorgs ewig leben. Ob dies tatsächlich passiert, ist nicht untersucht worden, aber es könnte der Grund dafür sein, daß ein künstlicher Alterungsmechanismus eingeführt wurde.

Damit ein Artorg sich vermehren kann, muß er genug Energie haben **und** ein vorher festgelegtes Alter überschreiten. Außerdem sind die Energiekosten altersabhängig ansteigend, so daß ein Artorg, der relativ alt ist, nicht mehr genug Futter sammeln kann, um seinen eigenen Energieverbrauch zu decken. Dies hat zur Folge, daß man die Altersverteilung nicht mehr wie in Anhang A abschätzen kann.

Zusammenfassung: Man erkennt, wie schwierig es ist, ein Modell zu bilden, das die Möglichkeit zur Evolution enthält, ohne gleichzeitig die Evolution genau festzulegen, wie es leider in diesem Modell passiert ist. Das Tauschen von Futter ist eine notwendige Voraussetzung für Kolonienbildung, gleichzeitig ist das Modell aber so spezifiziert, daß nur noch Kolonienbildung günstig ist. Die Artorgs haben keine andere Wahl.

Diese Arbeit [4] zeigt eine Tendenz, die leider auch noch in anderen Arbeiten aus dem Bereich Artificial Life zu finden ist. Es wird ein Phänomen gezeigt (hier die Kolonienbildung) und es wird angenommen, daß die im Laufe des Evolutionsprozesses erzeugten Individuen *fitter* sind als die Anfangspopulation, die zufällig erzeugt wurde. Dies ließe sich relativ leicht nachprüfen, wenn man folgende bisher nicht durchgeführten Tests machen würde:

1. Man löst eine Kolonie auf, indem die Artorgs gleichmäßig über das Gitter verteilt werden, und vergleicht die dann entstehende Dynamik, mit einer Dynamik von zufällig erzeugten Artorgs. Da keine Kolonie existiert, sollten sich beide Populationen ähnlich verhalten. Es könnte aber auch sein, daß die durch Evolution erzeugten Artorgs schneller Kolonien bilden, als zufällig erzeugte Artorgs.
2. Man ersetzt in einer Kolonie einen Teil der Artorgs durch solche mit einer zufälligen Tauschtable, diese sollten dann mit höherer Wahrscheinlichkeit sterben, da sie dem Leben in der Kolonie nicht angepaßt sind.

Ein weiterer Kritikpunkt an den beiden bisher vorgestellten Varianten besteht darin, daß die Bugs bzw. Artorgs schon mit relativ hoher Intelligenz ausgestattet sind. In der in Abschnitt 6.1 vorgestellten Version sind sie in der Lage, einen Futtergradienten festzustellen, und in der gerade diskutierten Version können sie das ihnen am nächsten befindliche Futterelement erkennen und sich darauf zu bewegen. Eine Abwandlung, die von *dümmere*n Bugs ausgeht, wird in [11] diskutiert und im nächsten Abschnitt vorgestellt. Dieses Modell dient dazu, den Begriff evolutionäre Aktivität einzuführen und zu erläutern.

6.3 Evolutionäre Aktivität

Evolution ist ein schwer zu definierender Begriff. Man hat eine intuitive Vorstellung, wann Evolution stattfindet, aber kein objektives, quantitatives Maß, das Evolution anzeigt. In thermodynamischen Systemen wird ein makroskopischer Zustand durch thermodynamische Variablen wie z.B. Temperatur, Druck und spezifische Wärme charakterisiert. Thermodynamische, makroskopische Variablen sind typischerweise statische Größen, die nur dann wohldefiniert sind, wenn zumindest *lokal* näherungsweise Gleichgewicht herrscht. Diese Variablen beziehen sich aber nur auf Freiheitsgrade, die vom Standpunkt der Evolution aus gesehen irrelevant sind, so daß man sich nach anderen für die Evolution relevanten Variablen umsehen muß.

Eine Biosphäre ändert sich ständig, so daß eine makroskopische Charakterisierung unvermeidlich dynamische Eigenschaften einschließen muß. Evolutionäre Dynamik besitzt häufig metastabile Zustände. Deshalb ist es sinnvoll, makroskopische Größen zu definieren, die Mittelwerte über kurze Zeiträume sind, und dann deren Änderung über längere Zeiträume zu verfolgen. Als eine solche Größe wird die *evolutionäre Aktivität* von BEDAU und PACKARD [11] eingeführt und auf das *Evolving-Bugs-Modell* angewendet. Ein Problem bei Evolutionsmodellen besteht darin, daß man a priori nicht wissen kann, welche Größen geeignet sind, die evolutionäre Aktivität in einem System zu beschreiben. Daher ist zu vermuten, daß *eine* Meßgröße nur ungenügende Informationen über den Evolutionsprozeß geben kann. Dies wird in den Abschnitten 6.3.3 und 6.3.4 noch genauer diskutiert.

Zunächst werden die Veränderungen des Bugsmodells im Vergleich zu Abschnitt 6.1 beschreiben, bevor dann am Beispiel dieses Modells die evolutionäre Aktivität erklärt wird, wie sie von BEDAU und PACKARD [11] definiert worden ist. Im Anschluß daran wird diese Definition mit Hilfe von Ergebnissen diskutiert, die aus eigenen Simulationen gewonnen wurden. Da Norman H. Packard freundlicherweise den Quelltext seines Simulationsprogramms zur Verfügung gestellt hat, konnten seine Simulationen nachvollzogen werden und außerdem eigene Meßroutinen eingebaut werden.

6.3.1 Strategische Bugs

Das Futter regnet in dieser Version nicht mehr in regelmäßigen CPU-Zeitabständen auf das Gitter, sondern es werden nun Futterpyramiden, die eine zentrale Höhe von 250 Einheiten und eine Breite von 25% der linearen Gittergröße haben, an zufälligen Orten auf das Gitter gebracht. In der hier besprochenen Version fällt eine Pyramide alle zwei Zeitschritte auf das Gitter.

Die beiden Gene, die die Schwelle für Nachwuchs und die Anzahl des Nachwuchses steuern, werden beibehalten, allerdings werden sie in den hier beschriebenen Experimenten nicht mutiert, sondern auf die festen Werte $g_{th} = 1000$ und $g_{off} = 2$ gesetzt. Die Bewegung eines Bugs ist nun nicht mehr für alle Bugs fest vorgegeben (in Richtung auf das meiste Futter in der Umgebung wie in Abschnitt 6.1), sondern nun folgt jeder Bug seiner eigenen Bewegungsstrategie.

Die Bewegungsstrategie eines Bugs ist eine Abbildung der lokalen Konfiguration des Futters auf einen Vektor, der die Richtung und Größe der Bewegung angibt. Ein Bug kann das Futter auf dem Platz, auf dem er sich befindet, und auf den vier Nachbarplätzen (dies ist die von Neumann-Umgebung) mit einer Auflösung von jeweils zwei Bits sehen (wenig, etwas mehr, noch mehr und viel Futter). Jeder dieser $2^{10} = 1024$ möglichen Umgebungen wird im Genom des Bugs (der Strategietabelle) ein Vektor zugeordnet, der besagt,

in welche Richtung (Stehenbleiben, Nord, Nordost, Ost, . . . , Nordwest, Zufall) sich der Bug wieviele Schritte (0 bis 15) bewegt. Damit hat ein Bug in diesem Modell nun 1024 Gene, von denen jedes 160 Allele hat (10 Richtungen mit 16 möglichen Schrittweiten). In der Anfangspopulation der Bugs ist die Strategietabelle so belegt, daß in 25% der Fälle in eine zufällige Richtung gegangen wird und in den restlichen 75% der Fälle der Bug stehenbleibt.

In diesem Abschnitt wird (in Anlehnung an den Originalartikel [11]) der Begriff *Genom* für die Strategietabelle verwendet. Dieser Begriff bietet sich an, wenn man an den in diesem Modell verwendeten Replikationsmechanismus denkt. Andererseits ist das Konzept der *Gene* eng mit der Unterscheidung zwischen Genotyp und Phänotyp verbunden, wie dies in Abbildung 3.1 in Kapitel 2 dargestellt wird. In diesem Modell dagegen ist das Genom, wie schon weiter oben beschrieben wurde, eine direkte Abbildung der Inputs (den Futterkonfigurationen) auf die Outputs (die Bewegungsdirektiven), die direkt zur Fitness (dem Energievorrat) beitragen. Wenn man an die Verwendung des Begriffs *Gen* im üblichen Sinne denkt, z.B. Gen für die Augenfarbe, so fällt einem in diesem Modell z.B. die bevorzugte Bewegungsrichtung als allgemeines Charakteristikum eines Bugs ein. Dies wird hier aber nicht durch ein einzelnes Gen vorgegeben, sondern nur durch die Verteilung der Allele im Genom. Auch wenn damit der Begriff *Gene* hier unpassend erscheint, wird er in diesem Kapitel weiterhin aus Bequemlichkeitsgründen verwendet.

Bei der Replikation wird ein Anteil (hier 10%) der Gene mutiert, außerdem kann auch noch sexuelle Reproduktion stattfinden, bei der Crossover zwischen den Genomen der *Eltern* vorgenommen wird. Wie schon in Abschnitt 6.1 gibt es auch wieder Energiekosten (8 Einheiten pro Schrittweite plus 26 Einheiten nur für das Überleben). Ein Bug kann pro Zeitschritt maximal 50 Einheiten Futter zu sich nehmen.

Alle übrigen relevanten Parameter wurden für eigene Simulationen aus [11] übernommen. Dabei ergab sich ein Problem bei der Anfangspopulation. Da 75% der unterscheidbaren Futterkonfigurationen dazu führen, daß der Bug stehen bleibt, stirbt ein Großteil der Anfangspopulation, bevor er Nachwuchs zeugen kann. Deshalb wurde bei eigenen Simulationen die Anfangspopulation von 50 auf 160 Bugs erhöht. Aber auch dies reichte noch nicht aus, um mit Sicherheit eine Simulation starten zu können. Daher wurde die Anfangsbelegung der Strategietabelle verändert. In 75% der Fälle geht der Bug in eine zufällige Richtung mit Schrittweite 1 und in nur 25% der Fälle bleibt er stehen. Eine Anfangspopulation von 160 Bugs ging dann immer noch in 140 Zeitschritten auf weniger als 10 Bugs zurück. Danach aber erholte sich die Bevölkerungszahl und pendelte sich schließlich nach etwa 5 000 Zeitschritten auf ungefähr 250 Bugs ein.

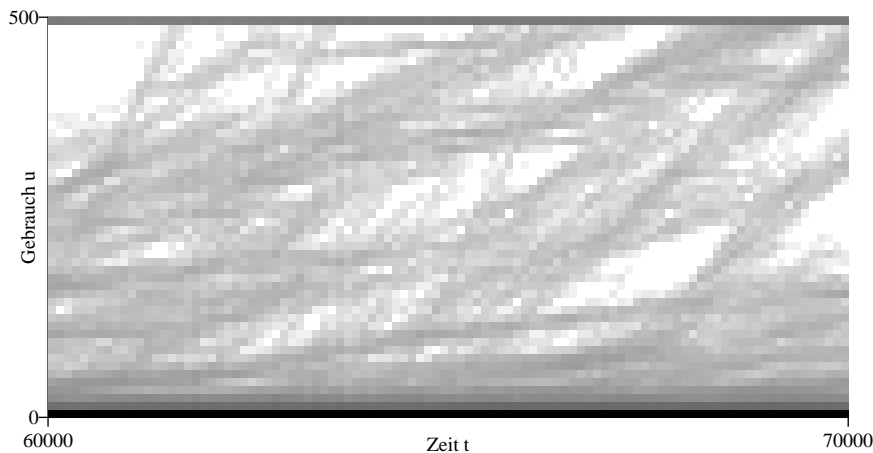
6.3.2 Aktivitätswellen

Ein sich aktiv entwickelndes System unterliegt ständigen Änderungen, an die es sich anpaßt. Sein Genpool verändert sich dauernd, dabei werden neue Gene aufgenommen und vorhandene Gene umarrangiert. Genetische Veränderungen können sich auf lange Sicht aber nur in der Bevölkerung halten, wenn die Organismen durch die neuen Gene gedeihen. Jede Veränderung, die die Fähigkeit zu überleben vermindert, wird auf lange Sicht aus der Bevölkerung eliminiert. Das bedeutet aber auch andererseits, daß diejenigen Veränderungen, die wirklich in den Genpool aufgenommen werden, das Überleben erleichtern oder im schlimmsten Fall neutral sind. Damit zeigt die ständige Aufnahme neuen, brauchbaren genetischen Materials, daß die Bevölkerung ständig ihren Genpool verbessert [11].

Die Rate, mit der neue Gene erzeugt werden, spiegelt nicht die wirkliche evolutionäre

Aktivität wieder, da diese neuen Gene nutzlos sein können. Genausowenig ist das bloße Vorhandensein genetischer Erneuerung allein relevant, da ein vorhandenes Gen nicht verwendet werden muß und daher irrelevant sein kann. Der *ständige Gebrauch* neuer Gene zeigt die wirkliche evolutionäre Aktivität. Im Modell der strategischen Bugs läßt sich das besonders leicht verstehen. Ein Gen wird gebraucht, wenn eine bestimmte lokale Futterkonfiguration (ein Input) vorgefunden wird. Die Aktion (der Output), die daraus resultiert, ist positiv und führt zu längerem Leben und mehr Nachkommen, wenn sie mehr Futter einbringt, als sie verbraucht. Ein Gen, das ständig verwendet wird, muß einen positiven Effekt haben, denn sonst würde es aus der Bevölkerung entfernt werden [11].

Abbildung 6.1: Aktivitätswellen im Evolving-Bugs-Modell



Diese Abbildung zeigt die in Gleichung (6.2) definierte Verteilungsfunktion als Grauwerte über der Zeit/Gebrauchs-Ebene. Die Grauwerte werden durch eine logarithmische Skala erzeugt, die die Aktivitätswellen besonders deutlich macht. In Zeit-Richtung fand eine Ausgabe alle 100 Zeitschritte statt. In Gebrauchs-Richtung wurden jeweils 10 Werte zu einem Eintrag zusammengefaßt.

Wie läßt sich nun der ständige Gebrauch der Gene messen? An jedem Gen j jedes Lebewesens i wird ein Zähler u_{ij} angebracht. Dieser *Gebrauchszähler* wird initialisiert, wenn ein Gen durch Mutation erzeugt wird, und er wird inkrementiert, wenn auf ein Gen zugegriffen wird. Er wird an die Nachkommen weitergegeben, wenn das Gen nicht durch Mutation verändert wird. Betrachtet man nun die Verteilungsfunktion

$$N(t, u) = \frac{1}{N_g} \sum_{i,j} \delta(u - u_{ij}^t), \quad (6.2)$$

die den Anteil der Gene an der Gesamtbevölkerung angibt, die bis zum Zeitpunkt t noch vorhanden sind und u -mal gebraucht wurden, so kann man Aktivitätswellen sehen (siehe Abbildung 6.1). Jede Welle entspricht einer Gruppe von Genen, die etwa gleich häufig in einem Zeitintervall verwendet werden. Betrachten wir nun Abbildung 6.1 etwas genauer. Man sieht, daß ein sehr großer Anteil der Gene nur sehr selten gebraucht wird. Dieser kurzzeitige Gebrauch ist von geringer Bedeutung. Sind diese Gene von Nachteil für den Organismus, werden sie schnell herausselektiert. Nur wenn der Gebrauch eines Gens eine Schwelle u_0 (diese beträgt in Abbildung 6.1 etwa 100 und ist sehr stark vom Modell abhängig) überschreitet, kann man davon ausgehen, daß dieses Gen nur aufgrund seiner Nützlichkeit so häufig verwendet wurde.

Bis zu diesem Punkt folgt die vorliegende Darstellung sehr stark den Ideen, die in [11] präsentiert werden. Die Hauptaussage läßt sich zusammenfassen mit dem Satz:

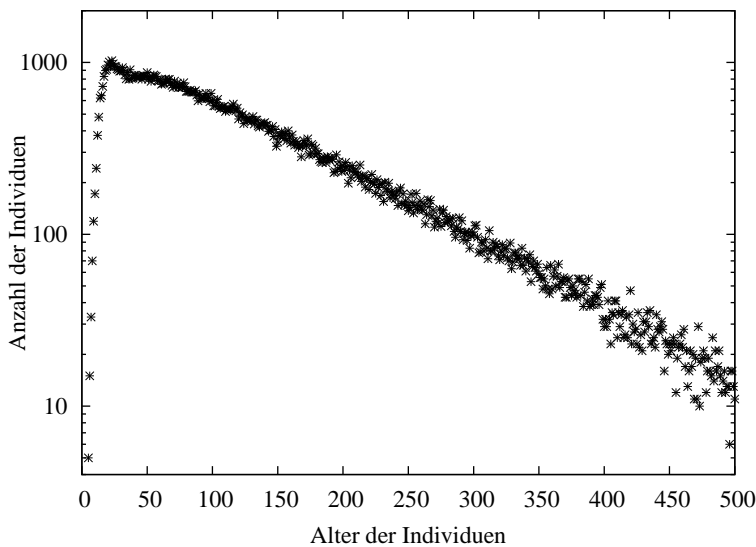
Nur nützliche Gene tragen zu den Aktivitätswellen bei.

Dies erscheint auf den ersten Blick plausibel, doch es kommen bei näherer Betrachtung einige Zweifel auf, die im folgenden Abschnitt bei der Diskussion eigener Simulationen erläutert werden sollen.

6.3.3 Auswertung von Simulationen

Die Altersverteilung: Damit ein Bug jeden Eintrag seiner Strategietabelle einmal testen kann, muß er mindestens 1024 Zeitschritte existieren. Wie man aber bei der Analyse der in Abbildung 6.2 dargestellten Daten ersehen kann, ist die durchschnittliche Lebenserwartung τ eines Bugs nur 129.2 Zeitschritte. (Die Größe τ läßt sich auch sehr leicht aus den in Anhang A definierten Größen der mittleren Bevölkerung pro Zeitschritt \bar{N} , der Simulationsdauer T und der Gesamtzahl der eliminierten Bugs N_d zu $\tau = \frac{T\bar{N}}{N_d} = 128.4$ berechnen). Ab $t_e \sim 70$ läßt sich die Verteilung sehr gut durch ein Exponentialgesetz

Abbildung 6.2: Altersverteilung der Bugs



Diese Abbildung zeigt die Altersverteilung der Bugs für das Modell von BEDAU und PACKARD für einen Lauf über $t_{sim} = 70\,000$ Zeitschritte bei einer Gittergröße von 128^2 Feldern. Man erkennt deutlich eine Exponentialverteilung, die ab $t_e \sim 70$ einsetzt und einer Eliminationswahrscheinlichkeit von $p = 9.11 \cdot 10^{-3}$ entspricht (siehe hierzu Anhang A). Die deutliche Abweichung von der Exponentialverteilung bis t_e läßt sich durch die spezielle Dynamik des Modells verstehen (siehe hierzu den Text).

beschreiben. Man sieht allerdings im vorderen Bereich der Verteilung bis t_e deutliche Abweichungen von dieser Verteilung. Diese Abweichungen lassen sich aber bei genauerer Analyse der Dynamik recht gut verstehen:

1. Jeder Bug startet mit etwa 500 Einheiten Futter. Damit kann er, falls er nur stehenbleibt, etwa 20 Zeitschritte überleben. Bugs, die durch Mutationen eine ungünstige Strategietabelle erhalten, werden also nach etwa 20 Zeitschritten eliminiert. Dies erklärt die Überhöhung der Verteilung bei $t = 20$.
2. Bei einer Mutation können alle der 160 möglichen Bewegungsdirektiven mit gleicher Wahrscheinlichkeit erzeugt werden. Findet ein Bug direkt nach seiner Erzeugung eine lokale Umgebung vor, die von ihm verlangt, einen Schritt der Schrittweite 15 zu machen, so verbraucht er in diesem Zeitschritt 146 Einheiten Energie. Dies geschieht mit sehr geringer Wahrscheinlichkeit in den ersten paar Zeitschritten seiner Existenz jedes Mal. Findet er in allen diesen Schritten am Zielfeld kein oder nur wenig Futter vor, so hat er nach 4 bis 5 Zeitschritten keine Energie mehr und wird eliminiert. Die Verteilung kann also erst ab $t \sim 5$ anfangen. In der Simulation, die Abbildung 6.2 zugrunde liegt, findet man dies bestätigt: Es gibt 5 Bugs, die nur 5 Zeitschritte überlebt haben.
3. Zwischen diesen beiden Punkten der Verteilung, dem Start bei $t = 5$ und dem Maximum bei $t \sim 20$, steigt die Verteilung monoton an. Man kann sich leicht überlegen, daß die Anzahl der möglichen Strategietabellen, die dazu führen, daß ein Bug, ohne Nachwuchs zu erzeugen, nach t Zeitschritten eliminiert wird, monoton mit t zunimmt. Dieser Teil der Verteilung läßt sich also gut verstehen, wenn man

davon ausgeht, daß er von Bugs gebildet wird, die sozusagen als *lebensuntüchtige* Individuen erzeugt werden.

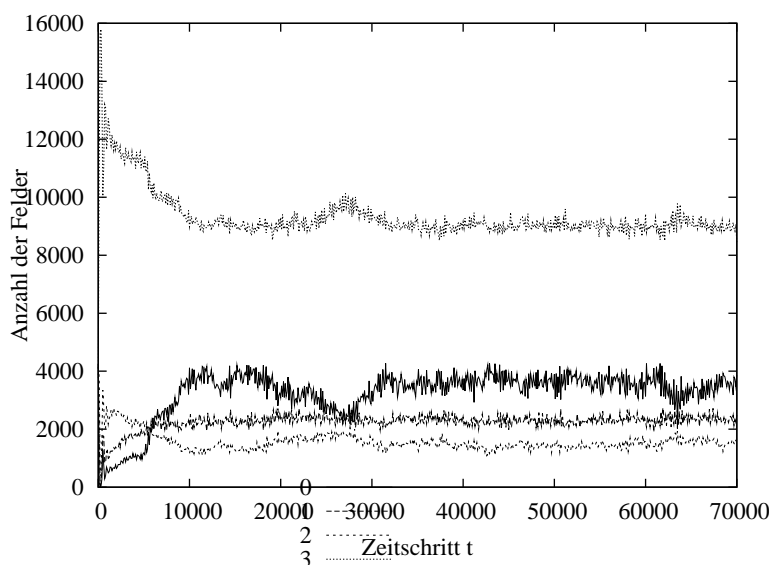
4. Pro Zeitschritt wird ein Anteil $\bar{p} = 7.78 \cdot 10^{-3}$ der Bevölkerung (siehe hierzu auch Anhang A) eliminiert. Wenn man davon ausgeht, daß bis zum Zeitpunkt t_1 kein Bug eliminiert wird und ab dann jeder Bug mit der Eliminationswahrscheinlichkeit p , so kann man unter Verwendung von Gleichung (A.13) t_1 zu

$$t_1 = \frac{1}{\bar{p}} - \frac{1}{p} \sim 18.8 \quad (6.3)$$

berechnen. Dieser Wert entspricht ziemlich genau dem Maximum der Altersverteilung.

5. Zwischen dem Maximum bei $t_1 \sim 20$ und dem Einsetzen der Exponentialverteilung bei t_e liegt eine geringfügige *Delle* im Vergleich zu einer reinen Exponentialverteilung vor. Dies ist leicht verständlich, wenn man berücksichtigt, daß ein Bug normalerweise (vergleiche auch Abbildung 6.3) einige Zeitschritte lang Futter findet, bevor er durch eine ungünstige Verhaltensweise verhungert, so daß er deutlich länger als 20 Zeitschritte überlebt.
6. Der Rest der Verteilung läßt sich recht gut durch eine Exponentialverteilung beschreiben. Wie man in Anhang A sehen kann, genügt dazu die Annahme, daß alle Bugs etwa gleich gut sind und daß, wegen des Kampfs um dieselben Ressourcen, ein konstanter Anteil $p \sim 9.11 \cdot 10^{-3}$ in jedem Zeitschritt ab dem Zeitpunkt t_1 eliminiert wird. Verwendet man die in diesem Abschnitt angegebenen Werte für t_1 und p , so kann man mit Gleichung (A.14) die mittlere Lebensdauer zu $\tau \simeq 128.6$ berechnen. Dieser Wert liegt genau zwischen den beiden Werten für τ , die zu Beginn dieses Abschnitts angegeben wurden.

Abbildung 6.3: Dynamik des Futterfeldes

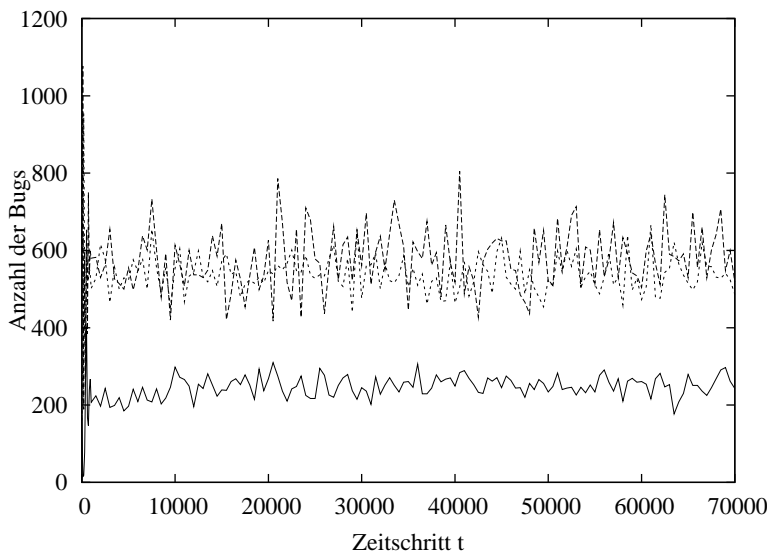


Diese Abbildung zeigt die Dynamik des Futterfeldes. Ein Bug kann, wie in Abschnitt 6.3.1 erläutert, jedes Feld mit einer Genauigkeit von 2 Bits untersuchen. Die Zahlen in der Legende der Abbildung geben an wieviel Futter mindestens enthalten ist. Die '3' bedeutet, das Feld ist mindestens zu 3/4 gefüllt, die '0' bedeutet, das Feld ist weniger als 1/4 gefüllt. Ein Feld kann maximal 255 Einheiten Futter bzw. Energie enthalten. Zu Beginn der Simulation ist jedes Feld zufällig mit einem Betrag zwischen 0 und 255 Einheiten belegt worden. Daher starten alle vier Kurven etwa bei 4000.

Das Futterfeld: Die Altersverteilung der Bugs zeigt, daß ein einzelner Bug nicht in der Lage ist, seine gesamte Strategietabelle zu testen. Da er aber bei der Reproduktion 90% der Strategietabelle unverändert an seinen Nachwuchs weitergibt, wird jedes einzelne Gen g über mehrere Generationen getestet und sein zugehöriger Gebrauchszähler $u(g)$ entsprechend inkrementiert. Da auf ein Gen nur zugegriffen wird, wenn die dazugehörige

Umgebung im Futterfeld realisiert ist, kann die Dynamik des Futterfeldes Aufschluß über den Anteil der Strategietabelle geben, der der Dynamik mit hoher Wahrscheinlichkeit zugänglich ist. Abbildung 6.3 zeigt eine solche Dynamik. Wie man auch an der durchgezogenen Linie aus Abbildung 6.4 deutlich erkennen kann, stirbt zu Beginn der Simulation ein Großteil der Anfangsbevölkerung aus. Da regelmäßig alle 2 Zeitschritte eine Futterpyramide auf das Gitter gebracht wird, führt das dazu, daß ein Großteil der Felder zu mehr als $3/4$ gefüllt wird. Die wenigen überlebenden Bugs haben dadurch Futter im Übermaß und können deshalb viel Nachwuchs erzeugen. Die Bevölkerung steigt zunächst auf über 400 Bugs an und pendelt sich dann zunächst mit abklingenden Oszillationen auf etwa 230 Bugs ein. Dies führt zu dem kleinen *Plateau* in den Kurven von Abbildung 6.3, das bis etwa $t = 4000$ verläuft. Nach etwa $t = 12000$ ist ein Gleichgewichtszustand erreicht, der bei etwa 250 Bugs liegt. Die Schwankungen um diesen Wert betragen etwa 15%. Dieser Gleichgewichtszustand wird bis zum Schluß der Simulation (eine Ausnahme bilden die Spitzen bei $t \sim 28000$) nicht mehr verlassen (siehe auch die durchgezogene Linie in Abbildung 6.4). Hieraus kann man nun schließen, daß ein Bug, selbst wenn er lang genug

Abbildung 6.4: Vergleich der Bevölkerungsgrößen



Diese Abbildung zeigt die Bevölkerungsgrößen von drei verschiedenen Simulationen. Die durchgezogene Linie entspricht dem Modell von BEDAU und PACKARD, die gestrichelte Linie gradient ascent und die gepunktete Linie den Bugs, die sich zufällig bewegen. Die Ausgabe der Bevölkerungsgröße fand alle 500 Zeitschritte statt, ohne daß eine Mittelung gemacht wurde, daher sind noch relativ starke Schwankungen vorhanden. Weitere Erläuterungen befinden sich im Text.

leben würde, gar nicht alle Futterkonfigurationen vorfinden kann. Nimmt man die Daten, die man aus Abbildung 6.3 ablesen kann, so haben etwa 3 der 5 untersuchten Nachbarfelder den Inhalt '3'. Betrachtet man alle Konfigurationen, die mindestens dreimal die '3' enthalten (dies sind 106 der 1024 möglichen Konfigurationen), so werden sie mit fast 60% Wahrscheinlichkeit vorgefunden.

Da jedes Gen mit gleicher Wahrscheinlichkeit bei der Reproduktion mutiert wird, lassen die bisherigen Ergebnisse folgendes Szenario wahrscheinlich erscheinen:

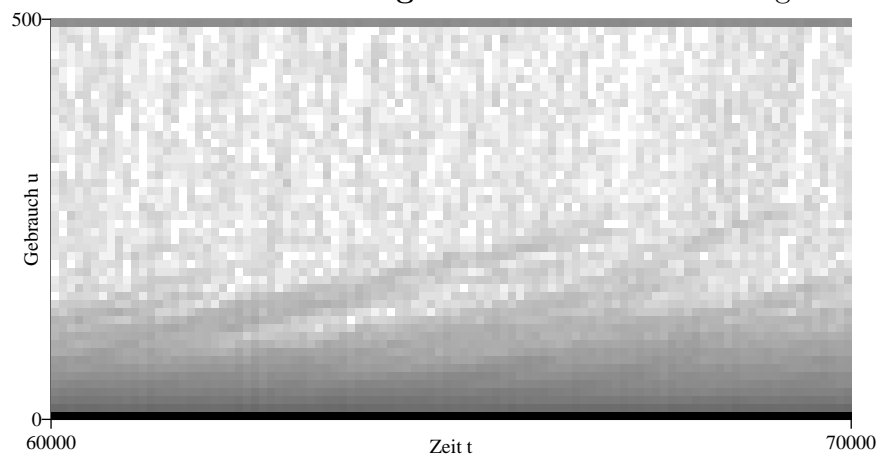
- Das Futterfeld, das Futter im Übermaß aufweist, ermöglicht in Verbindung mit der Lebensdauer der Bugs und der hohen Mutationsrate von 10%, daß ein geringer Anteil der Gene einen hohen Gebrauch $u(g)$ ansammeln, während die meisten Gene nur ganz geringen Gebrauch haben, bevor sie zerstört werden. Dies kann zu den beobachteten Aktivitätswellen führen, ohne daß evolutionäre Aktivität, wie sie von BEDAU und PACKARD definiert wird, vorliegt.

Die Voraussetzung: BEDAU und PACKARD gehen davon aus, daß eine Aktion eines Bugs einen eindeutigen Effekt hat. Sie ist entweder positiv oder negativ oder neutral. Dies ist aber in ihrem Modell **nicht** so realisiert. Eine Aktion kann zu einem Zeitpunkt t_a

positiv und zu einem anderen Zeitpunkt t_b negativ sein. Als Beispiel kann man die Aktion *Stehenbleiben* nehmen. Sitzt ein Bug im Zentrum einer Futterpyramide, so befinden sich dort 250 Futtereinheiten. Die direkt benachbarten Felder enthalten nur 125 Futtereinheiten. Da ein Bug nur 50 Einheiten pro Zeitschritt fressen kann, kann er 4 Zeitschritte stehenbleiben, in denen er jedesmal die maximale Futtermenge fressen kann, und spart so viermal die Energiekosten für eine Bewegung. Die Aktion *Stehenbleiben* ist also in diesem Fall eine positive Aktion. Sobald das Feld leergefressen ist, kann der Bug seinen Energieverbrauch nicht mehr decken, wenn er stehenbleibt, also ist die Aktion nun negativ. Eine ähnliche Argumentation läßt sich für jede andere Aktion ebenfalls konstruieren.

Dies unterstützt die obige Schlußfolgerung, daß die Aktivitätswellen nicht unbedingt ein Zeichen für evolutionäre Aktivität sein müssen. Die Gebrauchszähler $u(g)$ messen, welche Konfigurationen des Futterfeldes häufig vorgefunden werden. BEDAU und PACKARD gehen nun davon aus, daß erst die Kopplung dieser Inputgrößen an die Outputgrößen (die Aktion des Bugs) dafür sorgt, daß die charakteristischen Aktivitätswellen auftreten. Dies kann relativ einfach getestet werden. Man entfernt die Kopplung, indem die Aktion jedes Bugs unabhängig von der Futterfeldkonfiguration durchgeführt wird, ermittelt aber weiterhin die Statistik der Gebrauchszähler $u(g)$.

Abbildung 6.5: Aktivitätswellen bei gradient ascent

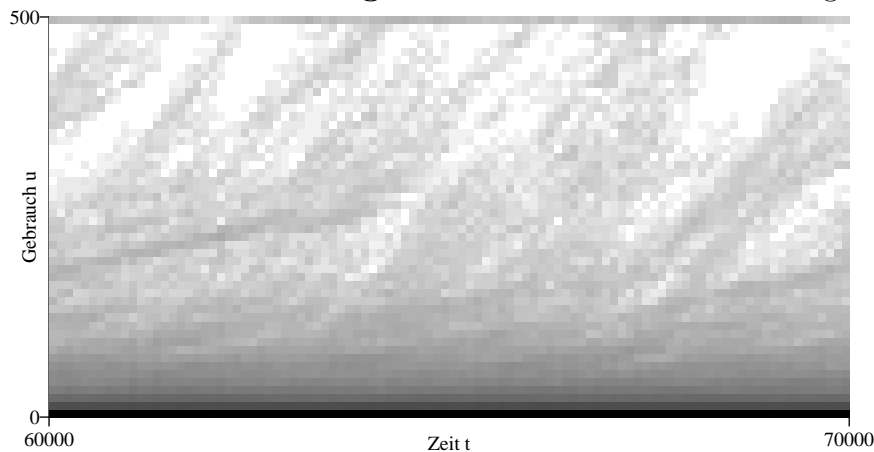


Diese Abbildung ist analog zu Abbildung 6.1. In diesem Fall geht ein Bug in das Nachbarfeld mit dem meisten Futter (*gradient ascent*). Dies entspricht der in Abschnitt 6.1 beschriebenen Dynamik.

Der Vergleich: Es wurden zwei verschiedene Tests durchgeführt. Abbildung 6.5 zeigt die Dynamik der Gebrauchszähler für den Fall, daß jeder Bug zu jedem Zeitschritt in das Nachbarfeld mit dem meisten Futter geht. Abbildung 6.6 dagegen zeigt den Fall, daß jeder Bug zufällig mit gleicher Wahrscheinlichkeit in eines der neun Nachbarfelder geht. Beiden Fällen ist gemeinsam, daß ein Bug stehenbleiben kann und dies auch garantiert tut, wenn das Zielfeld schon besetzt ist. Die Schrittweite beträgt aber grundsätzlich nur einen Schritt. Beide Abbildungen zeigen Aktivitätswellen.

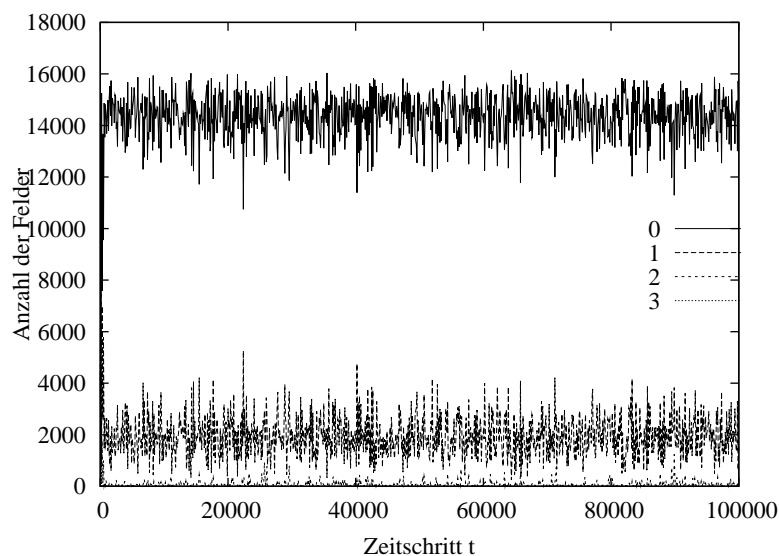
Eine Tatsache, die sofort ins Auge fällt, ist die deutlich höhere Bevölkerungsgröße (siehe Abbildung 6.4). Während sie bei dem Modell von BEDAU und PACKARD nur etwa 250 Bugs beträgt, liegt sie in beiden anderen Modellen bei über 500 Bugs. Diese Tatsache ist für die Bugs, die *gradient ascent* benutzen, sofort verständlich. Sie sammeln das Futter effektiver, wie die Dynamik des Futterfeldes in Abbildung 6.7 zeigt. Fast 90% aller Felder sind zu weniger als 1/4 gefüllt (siehe die durchgezogene Linie in Abbildung 6.7). Dies führt dazu, daß in diesem Fall ein Bug mit 94% Wahrscheinlichkeit eine Futterkonfiguration vorfindet, die aus mindestens dreimal '0' und höchstens zweimal '1' besteht. Dies sind nur 16 der möglichen 1024 Konfiguration. Hieraus kann man schließen, daß die Aktivitätswellen stärker ausgeprägt sein müßten und jede einzelne Welle sich durch eine stärkere

Abbildung 6.6: Aktivitätswellen bei zufälliger Bewegung



Diese Abbildung ist analog zu Abbildung 6.1. In diesem Fall geht ein Bug zufällig in eins der neun Nachbarfelder (der augenblickliche Standort zählt auch als Nachbarfeld). Die dort vorhandene Futtermenge ist dabei ohne Bedeutung.

Abbildung 6.7: Dynamik des Futterfeldes bei gradient ascent

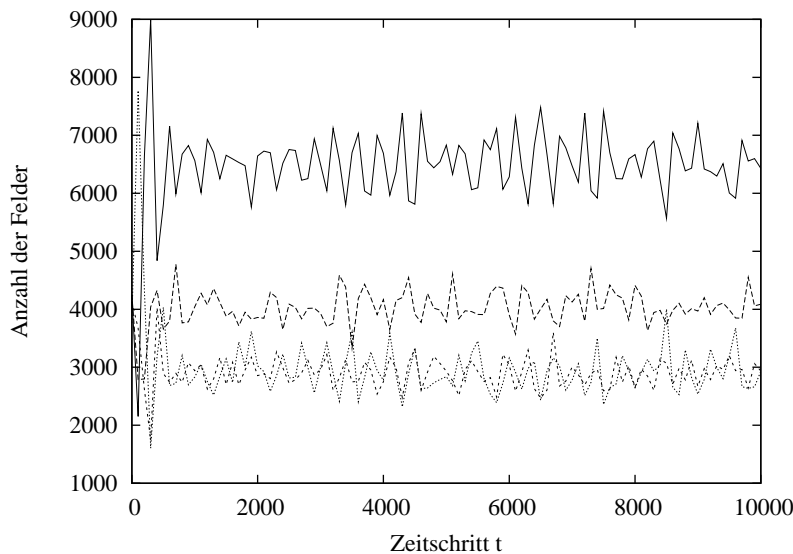


Diese Abbildung zeigt die Dynamik des Futterfeldes für Bugs, die gradient ascent verwenden. Die Parameter entsprechen im übrigen denen aus Abbildung 6.3.

Steigung auszeichnen sollte. Beides findet man in Abbildung 6.5 bestätigt. Man erkennt also, daß die Bugs, die gradient ascent benutzen, äußerst fähige Futtersammler sind. Fast die gesamte Energie, die von außen auf das Gitter gebracht wird, wird von den Bugs gefressen. Dies steht im Gegensatz zu dem Verhalten der Bugs, die die Strategietabelle benutzen.

Da ein Bug bei gradient ascent immer nur mit Schrittweite 1 geht, verbraucht er pro Zeitschritt nicht mehr als 34 Einheiten Energie. Ein Bug, der nach seiner Erzeugung keine Energie findet, kann so mindestens 15 Zeitschritte überleben. Dies wird durch die Altersverteilung bestätigt.

Bugs, die sich in zufällige Richtungen bewegen, sind nicht so effektive Futtersammler, wie Abbildung 6.8 zeigt, denn es befindet sich noch sehr viel Futter auf dem Gitter. Dennoch ist in beiden Fällen die mittlere Bevölkerungsgröße bei etwa 550 Bugs, wie man aus Abbildung 6.4 ablesen kann. Diese Tatsache läßt sich sehr leicht erklären, wenn man berücksichtigt, daß die treibende Kraft in beiden Fällen die Futterpyramiden sind, die alle 2 Zeitschritte auf das Gitter gebracht werden. Eine Futterpyramide kann maximal 36 000 Einheiten an Energie enthalten. Diese Zahl ist wirklich eine Obergrenze, denn überzähliges Futter wird entfernt, wenn ein Futterfeld mehr als 255 Einheiten enthält. Da jeder Bug zwischen 26 und 34 Einheiten pro Zeitschritt verbraucht, wird eine Populationsgröße von

Abbildung 6.8: Dynamik des Futterfeldes bei zufälliger Bewegung

Diese Abbildung zeigt die Dynamik des Futterfeldes für Bugs, die sich immer in eine zufällige Richtung bewegen. Die Parameter entsprechen im übrigen denen aus Abbildung 6.3 und 6.7.

etwa 550 Bugs ermöglicht. Ein Bug, der gradient ascent verwendet, wird häufiger stehenbleiben, da auch alle Nachbarfelder leergefressen sind (siehe Abbildung 6.7), so daß er im Schnitt etwas weniger Energie verbraucht. Daher beträgt die Bevölkerungsgröße 580 Bugs (siehe Abbildung 6.4). Bei den Bugs, die sich zufällig bewegen, ist mehr Futter auf dem Gitter (siehe Abbildung 6.8), so daß geringfügig weniger Energie durch die Pyramiden auf das Gitter gebracht werden kann, damit ergibt sich die etwas geringere durchschnittliche Bevölkerungsgröße von 540 Bugs. Damit ist auch verständlich, warum die Bugs, die die Strategietabelle verwenden, nur eine Bevölkerungsgröße von durchschnittlich 250 Bugs erreichen. Wie man aus Abbildung 6.3 erkennen kann, befindet sich immer sehr viel Futter auf dem Gitter, so daß eine Futterpyramide nur etwa zur Hälfte auf das Gitter gesetzt werden kann, der Rest läuft sozusagen über.

Die Dynamik des Futterfeldes bei Bugs, die sich in zufällige Richtungen bewegen, bewirkt, daß mehr Konfigurationen vorgefunden werden. Die 106 Konfigurationen, die mindestens dreimal die '0' enthalten, werden mit 33.5% Wahrscheinlichkeit vorgefunden. Bei den Bugs, die gradient ascent verwenden, lagen schon die 16 wahrscheinlichsten Konfigurationen mit etwa 94% Wahrscheinlichkeit vor. Dies hat zur Folge, daß die Aktivitätswellen bei zufälliger Bewegung eine wesentlich geringere Steigung haben, wie der Vergleich der Abbildungen 6.5 und 6.6 deutlich zeigt.

Resümee: Die in diesem Abschnitt vorgestellten Simulationen mit Zufallsmodellen zeigen eindeutig, daß die Aktivitätswellen keineswegs ausreichen, um evolutionäre Aktivität anzuzeigen. Dennoch findet im Modell von BEDAU und PACKARD Evolution statt. Um dies feststellen zu können, muß man allerdings andere Meßgrößen betrachten. Dies wird im folgenden letzten Abschnitt dieses Kapitels getan.

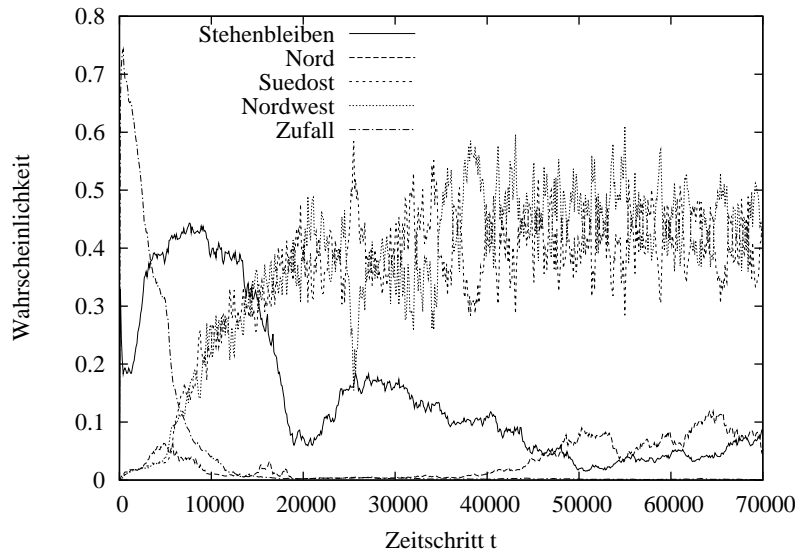
6.3.4 Nachweis von Evolution bei den strategischen Bugs

Aus den bisherigen Ausführungen über die Bugs von BEDAU und PACKARD, die eine Strategietabelle verwenden, ist klar geworden, daß nur ein geringer Teil der 1024 Gene häufig verwendet wird. Ihre Werte sollten also der Selektion unterliegen, da sie hohen Gebrauch ansammeln. Betrachtet man nur die Belegung der Strategietabellen, so sieht man, daß sich innerhalb von 10 000 Zeitschritten eine Gleichverteilung der Allele aus-

bildet. Mittelt man über die gesamte Bevölkerung, so wird jede der 10 Richtungen mit gleicher Wahrscheinlichkeit als Bewegungsdirektive vorgefunden und gleiches gilt auch für die 16 möglichen Schrittweiten. Wichtet man aber jeden Eintrag mit seinem Gebrauch und betrachtet man, welche Gene durch diese Allele vertreten werden, so ergeben sich ganz andere Verhältnisse. Die Schrittweite 1 liegt dann mit etwa 85% vor und die Schrittweite 0 (stehenbleiben) mit etwa 10%. Die übrigen Schrittweiten können vernachlässigt werden. Dieses Ergebnis ist leicht verständlich, denn es besagt, eine Bewegung mit der minimal möglichen Geschwindigkeit ist die optimale Bewegungsform.

Etwas anders sieht dies bei den bevorzugten Bewegungsrichtungen aus. Man kann noch relativ leicht verstehen, daß alle Bugs eine bevorzugte Richtung verwenden, z.B. Nord \leftrightarrow Süd, Ost \leftrightarrow West, Nordost \leftrightarrow Südwest oder Nordwest \leftrightarrow Südost. Auf diese Weise gehen alle Bugs parallel über das Gitter und können es ohne gegenseitige Störung *abgrasen*. Alle durchgeführten Simulationen zeigen aber, daß nur zwei der vier Möglichkeiten realisiert werden, und zwar nur die Fälle, in denen die Bugs diagonal über das Gitter gehen (Nordost \leftrightarrow Südwest und Nordwest \leftrightarrow Südost, in Abbildung 6.9 liegt der zweite Fall vor). Um dies verstehen zu können, muß man das genaue Aussehen einer Futterpyramide analysieren. Dabei erkennt man, daß das Gehen in Diagonalrichtungen dazu führt, daß der Bug sich in Richtung des stärksten Zuwachses bewegt. Dies ist daher günstiger. Die Bewegung in eine zufällige Richtung, die in der Anfangspopulation mit 75% Wahrscheinlichkeit ausgeführt wurde, ist nach etwa 10 000 Zeitschritten fast vollständig herausselektiert worden, obwohl 10% der Einträge aller Strategietabellen, noch diese Direktive geben.

Abbildung 6.9: Die bevorzugten Bewegungsrichtungen



Diese Abbildung zeigt die bevorzugten Bewegungsrichtungen für Bugs, die die Strategietabelle verwenden. Zu Beginn findet mit 75% Wahrscheinlichkeit eine Bewegung in eine zufällige Richtung statt und mit 25% Wahrscheinlichkeit bleibt der Bug stehen. Die Parameter entsprechen im übrigen denen aus Abbildung 6.3 und 6.7.

Mit den oben dargelegten Fakten erweist sich die Evolution auch in diesem Modell wieder nur als ein Optimierungsprozeß. Die Bugs fangen mit einer sehr ungünstigen Strategie an und innerhalb von 20 000 Zeitschritten wird eine bessere Strategie gefunden, die von da an als Gleichgewichtszustand vorliegt, um den herum nur noch Schwankungen stattfinden. Diese sind zwar wegen der hohen Mutationsrate von 10% relativ hoch, aber sie sind nicht hoch genug, um den Gleichgewichtszustand instabil zu machen.

Die in [11] vorgestellten Simulationen umfassen eine maximale Simulationszeit von $t_{sim} = 6000$ Zeitschritten. Wie die hier durchgeführten Simulationen zeigen, ist dies ein relativ kurzer Zeitraum. Die Optimierung der Strategietabelle ist erst nach 20 000 Zeitschritten abgeschlossen. Die in Abbildung 6.1 gezeigten Aktivitätswellen sind aber auch nach 60 000

Zeitschritten noch sichtbar.

Das Konzept der Gebrauchszähler erweist sich als interessant. Dieses allein reicht aber nicht aus, um evolutionäre Aktivität in einem Modell nachzuweisen, was durch Simulationen mit mehreren Zufallsmodellen zweifelsfrei festgestellt werden konnte. Damit zeigt sich, daß das Konzept der Zufallsmodelle, dem in dieser Arbeit besondere Aufmerksamkeit zukommt, ein sehr wichtiges Werkzeug liefert, um bei sogenannten Evolutionsmodellen, das Vorhandensein bzw. das Fehlen von Evolution anzuzeigen.

Dennoch ist die Untersuchung von solchen einfachen Modellen, wie es das Evolving-Bugs-Modell darstellt, sehr wichtig. Denn nur durch das Verständnis dieser simplen Modelle kann man hoffen, auch komplexere Systeme richtig analysieren zu können.

7 Emergenz

Ein in der Artificial Life Literatur häufig verwendeter Begriff ist die *Emergenz*. Ein Modell wird als gut spezifiziert angesehen, wenn es emergente Eigenschaften hat und emergentes Verhalten zeigt. Der Begriff *emergent* ist aber leider nicht eindeutig definiert, so daß er teilweise nur noch als Etikett verwendet wird, um ein Modell als erfolgreich zu kennzeichnen. Dieses Kapitel gibt nun keine neue und eindeutige Definition. Es soll in erster Linie dazu dienen, unterschiedliche Definitionen vorzustellen und einander gegenüberzustellen, so daß man Gemeinsamkeiten und Unterschiede besser erkennen kann.

Die Übersetzung des englischen Wortes *emergent*, wie man sie z.B. im *German-English, English-German Dictionary* von CASSELL [14] findet, lautet:

- **emergent** ... *adj.* auftauchend, aufsteigend, hervorgehend, emporkommend; (*fig.*) entspringend, entstehend, neu auftretend; ...

Damit hat man schon eine intuitive Vorstellung, was eine wissenschaftliche Definition des Begriffs alles umfassen sollte.

Bei den Wissenschaftlern, die sich mit Artificial Life beschäftigen, gibt es nur ein verschwommenes Konzept des Begriffs *Emergenz*, wie man einer Diskussion in der elektronischen Zeitschrift *Artificial Life Digest* [36] entnehmen kann. Es herrscht zwar Einigkeit darüber, daß Emergenz mit unterschiedlichen Ebenen der Beschreibung zu tun hat. Sobald dieses Konzept aber auf vorhandene Systeme angewendet werden soll, ergeben sich Probleme und es entsteht Ratlosigkeit. Im Folgenden werden einige Definitionen für den Begriff Emergenz vorgestellt. Die ersten beiden stammen aus den 60er Jahren von dem Philosophen NAGEL [95].

7.1 Zwei Definitionen von NAGEL

NAGEL beschreibt zwei verschiedene Konzepte von Emergenz [95]. Das erste bezieht sich auf Eigenschaften, die emergent im Sinne von unvorhersagbar sind; das zweite auf Eigenschaften, die im zeitlichen Sinne neu sind. Die erste Definition beschreibt also eine *logische Beziehung* zwischen Eigenschaften und Sätzen über Eigenschaften, während die zweite Definition eine rein *historische* Entwicklung betrachtet.

Zunächst sei der Punkt der *Unvorhersagbarkeit* genauer definiert [95]: Man betrachte eine Menge O , die aus den Elementen a_1, \dots, a_n besteht, die zueinander in der komplexen Beziehung R stehen. Die Eigenschaften der Elemente von O gehören zu den Klassen A_1, \dots, A_n . Man nimmt nun an, daß vollständiges Wissen über alle Elemente vorhanden ist und daß man alle Eigenschaften von echten Untermengen von O kennt. Man nimmt nun weiter an, daß O durch die Menge der Eigenschaften P charakterisiert wird. Wenn es mindestens eine Eigenschaft P_e in P gibt, die nicht vorhersagbar ist, obwohl vollständiges Wissen vorliegt, dann wird O als ein *emergentes Objekt* bezeichnet und P_e wird als *emergente Eigenschaft* von O bezeichnet.

Diese Definition muß näher erläutert werden [95]:

1. Das Wort *vorhersagbar* wird hier im Sinne von *herleitbar mit voller logischer Strenge* verwendet. Wenn man die Eigenschaft P_e zwar erraten, aber nicht beweisen kann, daß sie auftauchen muß, dann ist P_e immer noch eine emergente Eigenschaft.
2. Um genau zu sein, sind es nicht *Eigenschaften*, sondern *Sätze*, die herleitbar sind. Darüberhinaus können Sätze über Eigenschaften komplexer Objekte nur dann aus

Sätzen über ihre Komponenten hergeleitet werden, wenn es eine *Theorie* gibt, die die Komponenten und ihre Wechselwirkung beschreibt.

3. Eine Eigenschaft, die im Rahmen einer Theorie emergent ist, kann bei Verwendung einer anderen Theorie herleitbar sein, d.h., sie ist nicht emergent.

In [95] gibt es hierzu eine ausführliche Diskussion. Eine ähnliche Sichtweise vertritt CHURCHLAND in einem Artikel von 1985 [24].

Die zweite Definition wird als die Doktrin der emergenten Evolution bezeichnet. Sie unterscheidet sich von der Unvorhersagbarkeitsdefinition dadurch, daß eine Eigenschaft eines komplexen Objekts zwar emergent sein mag, weil sie nicht hergeleitet werden kann, daß sie aber im historischen Sinne nicht neu ist. Andererseits mag es Eigenschaften geben, die vorhersagbar sind, die aber bisher nicht beobachtet wurden, da die nötigen Randbedingungen bisher nicht gegeben waren. Damit sind diese Eigenschaften im historischen Sinne emergent.

Bevor die weiteren Definitionen von Emergenz vorgestellt werden, soll kurz auf die beiden Konzepte eingegangen werden. Der historische Emergenzbegriff verursacht keine Probleme. In der Biologie mag es zwar schwierig sein, emergente Evolution nachzuweisen. In Computerexperimenten dagegen bereitet es keine Probleme, da man die Entwicklung einer Simulation vollständig untersuchen kann.

Anders sieht das mit dem ersten Emergenzbegriff aus. Dieser Definition liegt die mechanistische Sichtweise zugrunde, daß alles mathematisch herleitbar ist, daß jedes Problem lösbar ist. Von diesen Ideen hat man sich in den Naturwissenschaften und der Mathematik lösen müssen. An dieser Stelle sei das sogenannte *Halting-Problem* [62] genannt, für das Turing bewiesen hat, daß es nicht entscheidbar ist. In der Physik sollte in diesem Zusammenhang auf die Chaosforschung verwiesen werden. Es existieren einfache und vollständig deterministische Systeme, deren Verhalten nicht vorhergesagt werden kann [107]. Dies hat unter anderem zur Folge, daß man trotz großer Rechnerleistung das Wetter nur auf wenige Tage voraussagen kann. Damit scheint die Voraussetzung von Nagel, daß vollständiges Wissen über ein System existiert, doch recht fragwürdig.

Im Folgenden werden einige weitere Definitionen vorgestellt, die von Artificial Life Forschern diskutiert werden, und mit den Definitionen von NAGEL verglichen.

7.2 Die Definition von LANGTON

LANGTON gibt in seiner Einleitung zur Proceedings der ersten Konferenz über *Artificial Life* [82] eine Anzahl wichtiger Charakteristiken von Artificial Life Modellen:

Although I think that none of us could have put it into words at the time, I think that many of us went away from that tumultuous interchange of ideas with a very similar vision, strongly based on themes such as *bottom-up* rather than *top-down* modeling, *local* rather than *global* control, *simple* rather than *complex* specifications, *emergent* rather than *prespecified* behavior, *population* rather than *individual* simulation, and so forth.

Emergentes Verhalten ist also eine Eigenschaft, die Artificial Life besonders kennzeichnet. LANGTON unterscheidet emergentes von vorherbestimmten Verhalten. Vergleicht man dies mit den Definitionen von NAGEL, so erkennt man, daß LANGTON die Unvorhersagbarkeitsdefinition im Sinn hat, denn *vorherbestimmt* entspricht *vorhersagbar* und *herleitbar*.

LANGTON gibt aber nicht an, wie man diese Definition auf vorhandene Modelle übertragen kann.

7.3 Die Definition von ASSAD und PACKARD

Es gibt noch andere Versuche, Emergenz zu definieren. Einer davon versucht eine Skala, mit der man Emergenz messen kann, einzuführen [4]. Diese Skala reicht von nicht-emergent über *schwach* und *stark* emergent bis zu maximal emergent (entnommen aus [4]):

- **Nicht emergent:** Das Verhalten ist sofort herleitbar, sobald man die zugrundeliegenden Regeln sieht.
- **Schwach emergent:** Das Verhalten ist im Nachhinein aus den Regeln herleitbar, aber erst *nachdem* man es beobachtet hat.
- ⋮
- **Stark emergent:** Das Verhalten ist theoretisch herleitbar, aber seine Erklärung ist äußerst schwierig.
- **Maximal emergent:** Das Verhalten kann nicht aus den Regeln hergeleitet werden.

Diese Skala ist sehr subjektiv. Was für den einen sofort herleitbar und damit nicht-emergent ist, kann der andere erst herleiten, nachdem er es beobachtet hat, was die Eigenschaft schwach emergent machen würde. Vergleicht man dies mit dem ersten Emergenzbegriff, wie er von NAGEL verwendet wird, so sieht man, daß bis auf den letzten Punkt alle anderen Punkte ein Verhalten als (theoretisch) herleitbar bezeichnen. Damit beschreiben die ersten Punkte nicht-emergentes Verhalten und nur der letzte Punkt bezeichnet emergentes Verhalten im Sinne von NAGEL.

7.4 Zusammenfassung

Die drei vorgestellten Definitionen zeigen nur einen Ausschnitt aus der stattfindenden Diskussion zu diesem Thema. In Ausgabe 49 vom *Artificial Life Digest* [36] findet man eine von CARIANI zusammengestellte Literaturliste zum Thema Emergenz. CARIANI selbst hat sich auch ausführlich mit dem Thema Emergenz befaßt, hier soll nur auf einen neueren Artikel, der in [84] erschienen ist, verwiesen werden [20]. In diesem Artikel bezweifelt er, daß es überhaupt möglich ist, Simulationen durchzuführen, die emergente Phänomene zeigen, da alle Regeln von vornherein vorgegeben sind. Daher fehlt die *Offenheit* biologischer Systeme, die erst Emergenz möglich macht.

Wie man sieht ist der Begriff *Emergenz* schwer faßbar, er steht damit auf derselben Stufe wie *Leben* und *Komplexität*. Dies sind zwei weitere Begriffe, die in diesem Zusammenhang ständig gebraucht werden, die aber bisher ebenfalls noch nicht zufriedenstellend definiert werden konnten [22, 52, 78]. Diese Begriffe stehen noch mit einem weiteren in Zusammenhang: *Selbstorganisation*. Findet man in einer Simulation Selbstorganisation, so wird dies häufig als emergentes Phänomen bezeichnet. Somit ergibt sich, daß *emergent* und *selbstorganisierend* zwei synonyme Begriffe sind, dies wurde auch schon in [119] erwähnt. Damit ist die Aufgabe, eine zufriedenstellende Definition für Emergenz zu finden, zwar noch nicht leichter geworden, aber man hat doch zumindest eine intuitiv verständliche Vorstellung des Begriffes.

8 Zusammenfassung

Die in dieser Arbeit vorgestellten Modelle wurden so einfach wie möglich konzipiert. Dies soll verhindern, daß ein beobachtetes komplexes Verhalten schon inhärent in den Regeln vorliegt und nicht erst durch die Dynamik erzeugt wird, wie dies bei vielen der in der Literatur vorkommenden Modelle der Fall zu sein scheint. Ein Problem besteht nun darin, daß, selbst wenn die Systeme komplexes Verhalten zeigen, dieses nicht festgestellt werden konnte. Eine Ursache dafür liegt in einer fehlenden, anerkannten Definition von Komplexität. Gleiches gilt für den Begriff Emergenz, der in zunehmendem Maße in der Artificial Life Literatur auftaucht. Bei der Suche nach solchen Definitionen konnte im Rahmen dieser Arbeit leider kein Durchbruch erzielt werden. Dennoch konnten auch ohne einen solchen fundamentalen Durchbruch sehr interessante Ergebnisse gewonnen werden.

Das Bit-String-Modell weist Ähnlichkeiten zu den Modellen von KAUFFMAN [69, 70, 71] und von DERRIDA, HIGGS und PELITI [27, 56, 57] auf. Im Unterschied zu diesen Modellen, in denen die einzelnen Bits selber schon als Gene aufgefaßt werden, ist im Bit-String-Modell eine zusätzliche Hierarchieebene eingeführt worden: Unterabschnitte des Bit-Strings werden als *Gene* aufgefaßt, sie bilden die in der Wechselwirkung aktiven Gruppen. Diese Hierarchie ermöglicht selbst bei fester Individuenlänge (in Bits) eine variable Anzahl der aktiven Gruppen. Während also beim NK-Modell von KAUFFMAN die Definition von Spezies völlig willkürlich eingeführt wurde (siehe speziell [71]), um Co-Evolution untersuchen zu können (alle Spezies enthalten gleich viele Bits/Gene), ergibt sich im Bit-String-Modell auf natürliche Weise das Konzept der Spezies. Da beim Bit-String-Modell der Fitnessbeitrag eines Genes nur von Genen in anderen Individuen abhängt, wird Co-Evolution im reinsten Sinne untersucht.

Die genetischen Operatoren greifen auf der unteren Ebene der Bits an, die Wechselwirkung findet dagegen auf der oberen Ebene der Gene statt. Damit ist eine erste Unterscheidung zwischen Genotyp und Phänotyp möglich. Diese kann sich natürlich noch nicht mit der Genotyp-Phänotyp-Relation in der Natur messen, allerdings existieren auch nur zwei Hierarchiestufen, während biologische Systeme eine Vielzahl davon besitzen.

Bei dem einfachen NK-Modell mit nur einer Spezies handelt es sich um ein reines Optimierungsproblem. Da die Fitnesslandschaft für große K sehr hügelig ist, ergibt sich normalerweise eine Dynamik, die durch eine Art Randomwalk von einem lokalen Maximum zum nächsten charakterisiert werden kann. Diese Struktur der Fitnesslandschaft ist von vielen Optimierungsproblemen her bekannt, sie macht das Aufsuchen des globalen Maximums so extrem schwierig. Im Unterschied dazu kann für das NK-Modell mit mehreren Spezies und in gleichem Maße für das Bit-String-Modell **keine** globale Fitnessfunktion vorgegeben werden. Damit sind diese Systeme näher an der Natur, da auch dort bestenfalls eine von der Population abhängige Fitness definiert werden kann (MAYNARD-SMITH [91] verwendet den Ausdruck *frequency dependent fitness*). Dadurch erst kann neben *Genetic Takeover*, das schon bei reinen Optimierungsproblemen gefunden wird, auch Bistabilität oder sogar Multistabilität gefunden werden. Das Bit-String-Modell zeigt erste ermutigende Resultate, die in diese Richtung weisen.

Bei der Diskussion der verschiedenen Dekodierschemata hat sich herausgestellt, daß Startcodons ein sehr wirkungsvolles Mittel sind, um den Einfluß von Mutationen so lokal wie möglich zu halten. Daher ist es nicht weiter verwunderlich, daß sie in der Natur ebenfalls realisiert sind [29, 75].

Bei der Diskussion der Simulationsergebnisse des Bit-String-Modells haben sich *Zufalls-*

modelle als sehr wertvolles Hilfsmittel herausgestellt. Sie basieren auf demselben Konzept wie die flache Fitnesslandschaft [27, 56, 57]. Sie erst ermöglichen es, den Einfluß der Selektion genau festzustellen. Da im Zufallsmodell die Bewertung nicht vom Geninhalt abhängt, können einfache statistische Überlegungen durchgeführt werden. In dieser Arbeit wurden analytische Rechnungen zur Altersverteilung sowohl für Individuen als auch für Spezies durchgeführt. Bei der Individuenaltersverteilung sind die Abweichungen von der theoretischen Verteilung nach Gleichung (A.1) vollständig verstanden. Die Selektion bewirkt nur eine geringfügige Deformierung des Exponentialgesetzes. Im Gegensatz dazu sieht man bei der Speziesaltersverteilung einen qualitativ anderen Verlauf. Statt des berechneten Exponentialgesetzes ist über weite Bereiche ein Potenzgesetz zu erkennen. Damit liegt ein System vor, das nicht durch nur eine Zeitskala, sondern durch eine ganze Reihe von effektiven Zeitskalen beschrieben wird.

Leider deuten Langzeitsimulationen an, daß sich in den meisten Fällen eine sehr stabile Phase ausbildet. Daraus kann geschlossen werden, daß das Bit-String-Modell in der vorliegenden Form noch kein minimales Modell für offene Evolution darstellt. Es existiert leider keine *natürliche* Erweiterung des Bit-String-Modells, die einen anderen Wechselwirkungsmechanismus benutzt, der auf mehr Eigenschaften einer offenen Evolution hoffen läßt. Man kann sich mehrere alternative Mechanismen vorstellen, diese sind aber meistens so kompliziert, daß die von LANGTON [83] (siehe auch die Diskussion in Kapitel 2 und in Abschnitt 3.1) geforderte Einfachheit des Modells verloren geht.

Die Verwendung eines Zufallsmodells als Referenzsystem ist nicht auf das Bit-String-Modell beschränkt. Die Diskussion des Evolving-Bugs-Modells zeigt, daß auch dort neue Erkenntnisse über die Dynamik des Modells erhalten werden konnten. Viele der in der Literatur untersuchten Evolutionsmodelle verwenden komplizierte Regeln. Es wäre nun interessant, durch Anwendung von Zufallsmodellen herauszufinden, welchen Einfluß die Regeln und welchen Einfluß die Selektion hat. Ein besonderes Problem ist es dabei, das geeignete Zufallsmodell zu entwickeln (siehe z.B. [117]).

Das Evolving-Bugs-Modell ist ohne explizite Wechselwirkung der Individuen definiert. Damit stellt es ein pures räumliches System dar. Wenn man die Dynamik solcher räumlicher Systeme versteht, kann man ableiten, welche Phänomene erst dadurch entstehen, daß ein System mit Wechselwirkung z.B. auf ein Gitter gesetzt wird. Dieses wäre sicher eine lohnende Erweiterung des Bit-String-Modells.

In diesem Zusammenhang kann man auch das Modell des Wanderers im Sumpf sehen. Obwohl die Wanderer sich gegenseitig nicht sehen können, sondern nur indirekt über ihre Eindrücke im Boden wechselwirken, findet eine sehr effektive Abstoßung statt. Diese Abstoßung sorgt bei einem einzelnen Wanderer nicht dafür, daß sein Weg selbstvermeidend wird, bei vielen Wanderern dagegen verhindert sie, daß zwei oder mehr auf demselben Gitterplatz stehen.

Das Modell des Wanderers ist noch in anderer Beziehung sehr interessant und hat dadurch auch Auswirkungen auf Evolutionsmodelle. In Evolutionsmodellen wird immer mit Rauschen gearbeitet. Die Ergebnisse des Bit-String-Modells zeigen ja auch, daß ein gewisses Mindestmaß an Rauschen vorliegen muß, sonst wird statt eines Modells mit fortschreitender Evolution nur ein Optimierungsprozeß betrachtet. Dieses Rauschen wird normalerweise von außen eingeführt. Das Modell des Wanderers zeigt nun, daß sehr einfache, vollständig deterministische Systeme existieren, deren statistisches Verhalten dem eines zufälligen Modells entspricht, denn es erzeugt Rauschen aus sich selbst heraus. Somit kann der *Red Queen Effekt* zu aperiodischem Verhalten führen. Dies könnte helfen, Evolutions-

modelle zu erfinden, die auch ohne von außen eingeführtes Rauschen ständig fortschreitende Evolution zeigen.

A Die Altersverteilung der Individuen

In diesem Anhang wird die Individuenaltersverteilung berechnet. Zunächst geht man davon aus, daß in jedem Zeitschritt im Mittel ein konstanter Anteil $p \in [0, 1]$ der Individuen eliminiert wird. Dies ist für ein Zufallsmodell sicher eine gerechtfertigte Annahme. Danach wird untersucht, wie sich die Ergebnisse ändern, wenn die Eliminationswahrscheinlichkeit p_t vom Alter t der Individuen abhängt. Dies ist eine sinnvolle Annahme z.B. für das *Energieflußmodell* von WOLTER [119] (siehe auch Abschnitt 3.6) und das *Evolving-Bugs-Modell* von PACKARD [11, 96] (siehe auch Kapitel 6) ist.

A.1 Konstante Eliminationswahrscheinlichkeit

Bei einer konstanten Eliminationswahrscheinlichkeit ergibt sich sofort eine Exponentialverteilung für die Sterbewahrscheinlichkeit d_t , die den Anteil der Lebewesen angibt, der nach exakt t Zeitschritten eliminiert wurde. Da ein Lebewesen $t - 1$ Zeitschritte überlebt haben muß, um dann im letzten Zeitschritt eliminiert werden zu können, erhält man

$$d_t = (1 - p)^{t-1}p. \quad (\text{A.1})$$

In einer Simulation, die nur mit einer endlichen Populationsgröße N und endlicher Simulationszeit T arbeiten kann, ist es sinnvoller, mit der Anzahl n_t der Individuen zu arbeiten, die nach t Zeitschritten eliminiert wurden. Die Gesamtzahl der eliminierten Individuen N_d ergibt sich zu

$$N_d = \sum_{t=1}^{\infty} n_t = pNT. \quad (\text{A.2})$$

Damit erhält man

$$n_t = N_d d_t = (1 - p)^{t-1} p^2 NT. \quad (\text{A.3})$$

Jetzt kann man das Alter t_{alt} ausrechnen, das im Mittel genau ein Individuum während der Simulationszeit erreicht hat (d.h., es gilt $n_{t_{alt}} \simeq 1$):

$$t_{alt} \simeq 1 - \frac{\ln(p^2 NT)}{\ln(1 - p)}, \quad (\text{A.4})$$

Dies ist nicht das maximale Alter, das in einer Simulation zu finden sein sollte, denn es gilt im Allgemeinen noch $\sum_{t > t_{alt}}^{\infty} n_t > 1.0$. Deshalb sei mit t_{max} das Alter bezeichnet, das in einem Zufallsmodell noch höchstens einmal überschritten wird:

$$t_{max} \simeq 1 - \frac{\ln(pNT)}{\ln(1 - p)}, \quad (\text{A.5})$$

so daß t_{max} die kleinste ganze Zahl ist, für die $\sum_{t > t_{max}}^{\infty} n_t \leq 1.0$ gilt.

Diese Beziehungen gelten auch dann, wenn die Bevölkerungsgröße Schwankungen unterliegt, wie das im Energieflußmodell von WOLTER [119] und im Evolving-Bugs-Modell von PACKARD [11, 96] vorkommt. Dort arbeitet man dann mit der mittleren Bevölkerungsgröße \bar{N} und der mittleren Eliminationswahrscheinlichkeit \bar{p} . Diese Näherung ist ausreichend gut, wenn die Schwankungen gering gegen die Mittelwerte sind. (Beim Energieflußmodell müssen zusätzlich die Parameter geeignet gewählt werden.)

Eine weitere relevante Größe, die auch in Anhang B wieder verwendet wird, ist das mittlere Alter τ , mit dem die Individuen sterben:

$$\tau = \bar{t} = \sum_{t=1}^{\infty} d_t t. \quad (\text{A.6})$$

Für den Fall einer konstanten Eliminationswahrscheinlichkeit ergibt sich sofort durch Einsetzen von Gleichung (A.1) in Gleichung (A.6):

$$\tau = p \sum_{t=1}^{\infty} (1-p)^{t-1} t = \frac{1}{p}. \quad (\text{A.7})$$

A.2 Altersabhängige Eliminationswahrscheinlichkeit

In diesem Abschnitt wird vorausgesetzt, daß für $t \leq t_1$ eine altersabhängige Eliminationswahrscheinlichkeit $p_t = g(t) \leq p$ und für große t -Werte wieder eine konstante Eliminationswahrscheinlichkeit p vorliegt:

$$p_t = \begin{cases} g(t) & \text{für } t \leq t_1 \\ p & \text{für } t > t_1, \end{cases} \quad (\text{A.8})$$

so daß sich asymptotisch wieder eine Exponentialverteilung ergibt. Man erhält nun in Analogie zum vorigen Abschnitt:

$$d_t = \prod_{s=1}^{t-1} (1-p_s) \cdot p_t. \quad (\text{A.9})$$

Für die mittlere Eliminationswahrscheinlichkeit \bar{p} gilt dabei

$$\bar{p} = \sum_{t=1}^{\infty} p_t a_t = a_1, \quad (\text{A.10})$$

wobei $\{a_t\}$ die Wahrscheinlichkeitsverteilung des Alters der Individuen zu einem beliebigen Zeitpunkt bezeichnet, für die $d_t = \frac{a_t p_t}{a_1}$ gilt. Die Gleichung $\bar{p} = a_1$ spiegelt dabei die Tatsache wieder, daß bei einer etwa konstanten Bevölkerungsgröße der Anteil der eliminierten Individuen identisch mit dem Anteil der neu erzeugten Individuen ist. Es gilt

$$a_t = \prod_{s=1}^{t-1} (1-p_s) \cdot a_1, \quad (\text{A.11})$$

so daß man a_1 sehr leicht aus der Normierungsbedingung $\sum a_t = 1$ erhalten kann.

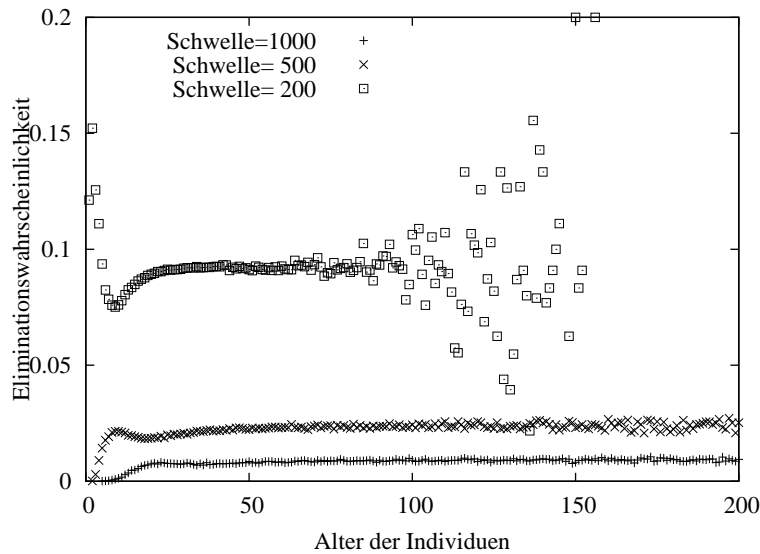
Betrachtet man den einfachen Fall $g(t) \equiv 0$, so erhält man nach kurzen Rechnungen die folgenden Gleichungen:

$$d_t = \begin{cases} 0 & \text{für } t \leq t_1 \\ p(1-p)^{t-(t_1+1)} & \text{für } t > t_1, \end{cases} \quad (\text{A.12})$$

$$\bar{p} = \frac{p}{1+pt_1} \quad (\text{A.13})$$

$$\tau = t_1 + \frac{1}{p}. \quad (\text{A.14})$$

Abbildung A.1: Eliminationswahrscheinlichkeiten im Bugsmodell



Diese

Abbildung zeigt die Eliminationswahrscheinlichkeiten der Bugs für das Modell von BEDAU und PACKARD [11] (siehe hierzu Kapitel 6 und speziell Abschnitt 6.3.1). Im Unterschied zu den Simulationen aus Kapitel 6 wurden hier drei verschiedene Reproduktionsschwellen g_{th} untersucht (diese Größe ist in Abschnitt 6.1 definiert worden). (Weiteres steht im Text.)

Für $t_1 = 0$ erhält man wieder die Formeln aus Abschnitt A.1. Normalerweise findet man in Simulationen kompliziertere Funktionen $g(t)$, aber der Fall $g(t) \equiv 0$ ist sehr oft eine gute erste Näherung. Dies sieht man sehr schön, wenn man Daten für das Evolving-Bugs-Modell, das in Kapitel 6 und speziell in Abschnitt 6.3.1 vorgestellt wurde, untersucht. In Abbildung A.1 sieht man die Eliminationswahrscheinlichkeiten für Simulationen mit drei verschiedenen Reproduktionsschwellen g_{th} . Diese Größe, die in Abschnitt 6.1 definiert wurde, regelt die Zeitskala und damit auch die Eliminationswahrscheinlichkeit. Man erkennt, daß Gleichung (A.8) mit $g(t) \equiv 0$ die Daten in erster Näherung recht gut beschreibt. Die Größe t_1 ist dabei natürlich ebenfalls von g_{th} abhängig. Die Werte von p und \bar{p} aus den Simulationen wurden verwendet, um dann mit den Gleichungen (A.13) und (A.14) τ zu berechnen. Dies liefert für $g_{th} = 500$ bzw. 1 000 Werte von $\tau = 49$ bzw. 128.5 im Vergleich zu den gemessenen Werten $\tau = 50$ bzw. 129, die aus der Altersverteilung $\{d_t\}$ stammen. Für $g_{th} = 200$ funktioniert diese Rechnung nicht, was durch folgende Argumentation verständlich wird: Wenn man Abbildung A.1 genau betrachtet, sieht man, daß für kleine Werte von t zunächst ein Maximum und dann ein Minimum vorliegt, bevor der nahezu konstante Wert $p = 9.16 \cdot 10^{-2}$ erreicht wird. In diesem Fall arbeitet man also besser mit

$$p_t = \begin{cases} p' & \text{für } t \leq t_1 \\ p & \text{für } t > t_1, \end{cases} \quad (\text{A.15})$$

wobei hier speziell $p' > p$ gilt. Dies führt zu

$$\bar{p} = \frac{pp'}{p + (p' - p)(1 - p')^{t_1}} \quad (\text{A.16})$$

$$\tau = \frac{1}{p'} + \left(\frac{1}{p} - \frac{1}{p'} \right) (1 - p')^{t_1} = \frac{1}{\bar{p}}. \quad (\text{A.17})$$

Hieraus erhält man nun $\tau = 9.363$ im Vergleich zum gemessenen Wert von $\tau = 9.167$ aus $\{d_t\}$.

Man erkennt deutlich, daß die einfache Annahme von stückweise konstanter Eliminationswahrscheinlichkeit zu recht guten Voraussagen führt. Die Abweichungen, die man in Abbildung A.1 sieht, können bei genauer Betrachtung der Dynamik (siehe Abschnitt 6.3.3) ebenfalls erklärt werden, so daß die in diesem Anhang durchgeführten Rechnungen im Rahmen dieser Arbeit ausreichen.

B Verzweigungsprozesse

B.1 Kurzer historischer Überblick

Im Rahmen dieser Arbeit wurde schon mehrfach darauf hingewiesen, wie wichtig sogenannte *Zufallsmodelle* sind. Dies hat schon 1873 GALTON [44] erkannt, als er sich mit dem Phänomen des Aussterbens von Familiennamen beschäftigte. Er wollte nicht unkritisch die vorher aufgestellte Hypothese akzeptieren, daß *vornehme* Familien mit höherer Wahrscheinlichkeit aussterben, als *gewöhnliche* Familien. Daher betrachtete er die Berechnung der Wahrscheinlichkeit des Aussterbens einer gewöhnlichen Familie aus den Fruchtbarkeitsdaten der gesamten Bevölkerung als ersten Schritt zur Lösung des Problems (nach HARRIS [55]). Die erste (leider falsche bzw. nicht vollständige) Lösung wurde schon im selben Jahr von WATSON [116] geliefert. Er verwendete erzeugende Funktionen, um zu zeigen, daß jeder Familienname ausstirbt. Nach diesen ersten Papieren tauchte das Problem erst in den 20er Jahren dieses Jahrhundert u.a. im Zusammenhang mit genetischen Forschungen wieder auf (siehe hierzu FISHER [34, 35], HALDANE [54] und STEFFENSEN [111, 112]). Eine davon unabhängige Wiederentdeckung des Problems fand im Jahre 1929 durch ERLANG [32] statt, der bemerkte, daß eine zweite Lösung möglich ist. Diese erlaubt das ewige Weiterbestehen eines Familiennamens unter der Bedingung, daß die durchschnittliche Zahl männlicher Nachkommen größer als 1 ist. Im Folgenden schlug ELDERTON eine Wahrscheinlichkeitsverteilung für die Zahl der Nachkommen vor, die es erlaubte, einfache analytische Ausdrücke zu finden. Diese verwendete LOTKA [86, 87], um sie auf amerikanische Zensusdaten anzuwenden, die dadurch sehr gut beschrieben werden konnten. Der Begriff *Verzweigungsprozeß* wurde erst 1947 in einem Papier von KOLMOGOROV und DMITRIEV [77] eingeführt.

Man kann sich Verzweigungsprozesse als eine mathematische Repräsentation der Entwicklung einer Bevölkerung vorstellen, deren Mitglieder sich fortpflanzen und sterben rein nach den Gesetzen des Zufalls. Die Objekte können unterschiedliche Eigenschaften haben, wie z.B. Alter, Energie, Position, oder auch andere Faktoren, sie dürfen aber nicht miteinander wechselwirken. Diese Annahme, die der mathematischen Theorie zugrunde liegt, scheint gerechtfertigt für Populationen einiger physikalischer Teilchen wie Neutronen oder kosmische Strahlung, sie erscheint aber nur von bedingter Relevanz für biologische Populationen (nach [55]).

B.2 Der Galton-Watson-Prozeß

Der originale Galton-Watson-Prozeß beschäftigt sich mit der Beantwortung der Frage [55]:

Wenn man die jeweiligen Wahrscheinlichkeiten p_0, p_1, p_2, \dots hat, daß ein Mann $0, 1, 2, \dots$ Söhne bekommt, von denen jeder dieselben Wahrscheinlichkeiten hat, selber Söhne zu bekommen, und so weiter, wie groß ist die Wahrscheinlichkeit, daß die Familie ausstirbt, und allgemeiner, wie groß ist die Wahrscheinlichkeit für eine gegebene Anzahl männlicher Nachkommen in einer gegebenen Generation?

Watson verwendete zur Lösung des Problems erzeugende Funktionen: Sei $\{p_k\}$ die Wahrscheinlichkeitsverteilung der Nachkommen, dann ist

$$f(s) = \sum_{k=0}^{\infty} p_k s^k \quad \text{mit } 0 \leq s \leq 1 \quad (\text{B.1})$$

die erzeugende Funktion der Wahrscheinlichkeiten. Außerdem seien die Iterierten der Erzeugenden durch

$$f_1(s) = f(s) \quad \text{und} \quad f_{n+1}(s) = f(f_n(s)) = f_n(f(s)) \quad \text{für} \quad n = 1, 2, \dots \quad (\text{B.2})$$

definiert. Dann hat die Potenzreihe für f_n als Koeffizienten $p_{n,k}$ die Terme der Wahrscheinlichkeitsverteilung für die Gesamtzahl der Männer in der n -ten Generation (der erste Träger des Namens entspricht der 0-ten Generation). Watson bemerkte schon, daß die Wahrscheinlichkeit q_n der Auslöschung bis zur n -ten Generation durch die Gleichungen

$$\begin{aligned} q_1 &= p_{1,0} = f(0) = p_0 \\ q_{n+1} &= p_{n+1,0} = f_{n+1}(0) = f(f_n(0)) = f(q_n) \end{aligned} \quad (\text{B.3})$$

beschrieben werden konnte. Wenn gilt $q_n \rightarrow q$ für $n \rightarrow \infty$, dann gilt $q = f(q)$. Diese Gleichung hat die Wurzel $q = 1$. Diese ist die einzige Wurzel im Intervall $[0, 1]$, wenn $m \leq 1$ mit

$$m = \sum_{k=0}^{\infty} k p_k = f'(1) = \text{mittlere Anzahl männlicher Nachkommen} \quad (\text{B.4})$$

gilt. Für $m > 1$ existiert aber eine weitere Lösung $q < 1$.

Wenn man davon ausgeht, daß die Wahrscheinlichkeiten p_1, p_2, \dots eine geometrische Reihe der Form $p_k = bc^{k-1}$ mit $k = 1, 2, \dots$, $0 < b, c$ und $b < 1 - c$ bilden, dann gilt erstens $p_0 = 1 - \sum_{k=1}^{\infty} p_k = 1 - \frac{b}{1-c}$ und zweitens ist $f(s)$ eine gebrochene rationale Funktion:

$$f(s) = 1 - \frac{b}{1-c} + \frac{bs}{1-cs} = p_0 + \frac{bs}{1-cs}. \quad (\text{B.5})$$

Für die mittlere Zahl männlicher Nachkommen ergibt sich somit $m = \frac{b}{(1-c)^2}$. Die Gleichung $s = f(s)$ hat außer $s = 1$ die positive Wurzel

$$s_0 = \frac{1-b-c}{c(1-c)} \begin{cases} > 1 & \text{für } m < 1, \\ = 1 & \text{für } m = 1, \\ < 1 & \text{für } m > 1. \end{cases} \quad (\text{B.6})$$

Durch Induktion kann man zeigen, daß $f_n(s)$ dieselbe Form mit anderen Koeffizienten b_n und c_n hat. Es gilt für $m \neq 1$

$$f_n(s) = 1 - m^n \left(\frac{1-s_0}{m^n - s_0} \right) + \frac{m^n \left(\frac{1-s_0}{m^n - s_0} \right)^2 s}{1 - \left(\frac{m^n - 1}{m^n - s_0} \right) s}, \quad n = 1, 2, \dots \quad (\text{B.7})$$

und für $m = 1$

$$f_n(s) = \frac{nc - (nc + c - 1)s}{1 - c + nc - ncs}, \quad 0 < c < 1; \quad n = 1, 2, \dots \quad (\text{B.8})$$

Diese Gleichungen reichen aus, um die Rechnungen des nächsten Abschnittes nachvollziehen zu können.

B.3 Anwendung auf das Bit-String-Modell

Zunächst wird gezeigt, daß die in dieser Arbeit verwendete Dynamik tatsächlich zu einer geometrischen Reihe für die Nachwuchsverteilung führt. Damit liegt es nahe, die Theorie der Verzweigungsprozesse zu verwenden, um die Altersverteilung der Spezies zu berechnen.

In jedem Zeitschritt wird ein Anteil P_0 der Individuen eliminiert, ein Anteil P_1 wird unverändert in den nächsten Zeitschritt übernommen und ein Anteil P_2 gelangt mit einem Individuum Nachwuchs in den nächsten Zeitschritt. Dabei gilt $P_0 + P_1 + P_2 = 1$ und $m = P_1 + 2P_2 = 1$ (d.h., $P_0 = P_2$), da mit konstanter Bevölkerungsgröße gearbeitet wird. Damit läßt sich, wie in Anhang A gezeigt wird, die Altersverteilung der Individuen sehr genau (nicht nur für reine Zufallsmodelle) berechnen. Für die Berechnungen in diesem Anhang muß allerdings berücksichtigt werden, daß nur solcher Nachwuchs gezählt wird, der zur selben Spezies wie der Vorfahr gehört. Bei einer Punktmutationsrate von $p_{mut} = 10^{-3}$ und einer Lebewesenlänge von 96 Bits ergibt sich eine Wahrscheinlichkeit M für identischen Nachwuchs von $M = (1 - p_{mut})^{96} \sim 0.91$, so daß nur noch ein Anteil $P'_2 = MP_2$ der Individuen mit Nachwuchs in die nächste Generation gelangt. (Der in Simulationen gemessene Wert für M liegt üblicherweise etwas höher, da durch Verwendung von Startcodons eine Mutation in einer Junkregion des Individuums dieses unverändert läßt.) Dies verändert natürlich auch den Anteil P'_1 der Individuen, die unverändert in den nächsten Zeitschritt übernommen werden. Zusammenfassend ergibt sich für die P'_i :

$$\begin{aligned} P'_0 &= P_0 \\ P'_1 &= 1 - (1 + M)P_0 \\ P'_2 &= MP_0. \end{aligned} \tag{B.9}$$

Hiermit können wir nun die $\{p_k\}$ für den Galton-Watson-Prozeß berechnen. Die Wahrscheinlichkeit p_0 für ein Individuum ohne Nachwuchs setzt sich zusammen aus der Wahrscheinlichkeit P'_0 , daß das Individuum im ersten Zeitschritt eliminiert wird, und aus der Wahrscheinlichkeit $P'_1 p_0$, daß das Individuum den ersten Zeitschritt ohne Nachwuchs überlebt und dann später ohne Nachwuchs eliminiert wird:

$$p_0 = P'_0 + P'_1 p_0. \tag{B.10}$$

Dies ergibt

$$p_0 = \frac{1}{1 + M}. \tag{B.11}$$

Analog läßt sich p_k aus p_{k-1} berechnen:

$$p_k = P'_1 p_k + P'_2 p_{k-1}, \tag{B.12}$$

so daß sich nach kurzer Rechnung für b und c , wie sie in Abschnitt B.2 definiert sind, folgende Formeln

$$b = \frac{M}{(1 + M)^2} \tag{B.13}$$

$$c = \frac{M}{1 + M} \tag{B.14}$$

ergeben.

Damit läßt sich nun Gleichung (B.7) verwenden, um die Wahrscheinlichkeiten Q_n auszurechnen, daß eine Spezies nach exakt n Zeitschritten ausstirbt:

$$Q_n = q_n - q_{n-1} = f_n(0) - f_{n-1}(0) \quad (\text{B.15})$$

$$= \frac{s_0(1 - M^n)}{s_0 - M^n} - \frac{s_0(1 - M^{n-1})}{s_0 - M^{n-1}} \quad (\text{B.16})$$

$$= \frac{s_0(1 - s_0)M^n(1 - \frac{1}{M})}{s_0^2 - (M^n + M^{n-1})s_0 + M^{2n-1}} \quad (\text{B.17})$$

$$= \frac{M^n(1 - M)^2}{(1 - M^n)(1 - M^{n+1})}, \quad (\text{B.18})$$

dabei ist $s_0 = \frac{1}{M}$ verwendet worden, das sich durch Einsetzen der Gleichungen (B.13) und (B.14) in Gleichung (B.6) ergibt. Man sieht also, daß man asymptotisch eine Exponentialverteilung erhält. Nebenbei sei an dieser Stelle noch bemerkt, daß sich für $M = 1$ aus Gleichung (B.8) eine Potenzverteilung berechnen läßt:

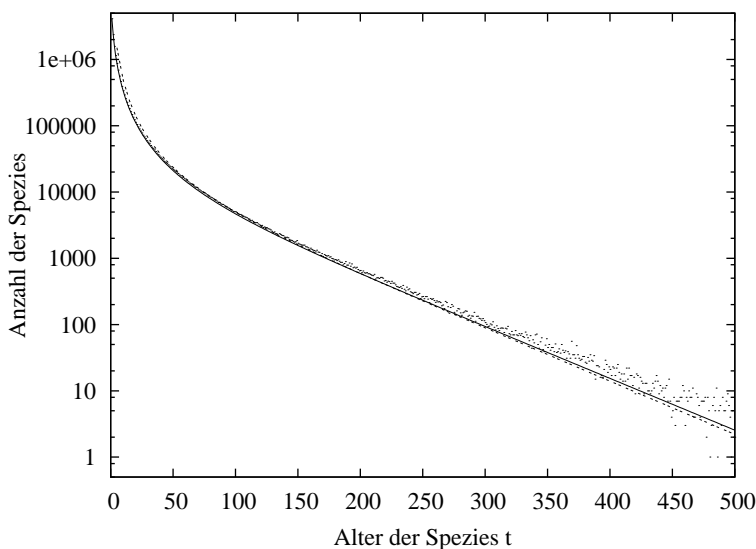
$$Q_n^{M=1} = \frac{1}{n(n+1)}. \quad (\text{B.19})$$

Bei der bisherigen Herleitung wurde unterstellt, daß die Dynamik in den Zufallsmodellen rein durch einen Galton-Watson-Prozeß beschrieben werden könnte. Dies ist natürlich nicht der Fall, denn in einem Galton-Watson-Prozeß wird u.a. von getrennten, nicht überlappenden Generationen ausgegangen, was auch für menschliche Generationen eine unrealistische Annahme ist. Außerdem wird die Zeit in Generationen und nicht in Zeitschritten gemessen. Dieser zweite Punkt bereitet keine Probleme, wenn man die gewonnene Gleichung (B.18) einfach von Generationen auf Zeitschritte unter Verwendung der mittleren Lebensdauer τ aus Anhang A umskaliert:

$$Q(t) = \frac{1}{\tau} Q_{t/\tau}. \quad (\text{B.20})$$

Diese Gleichung beschreibt die Daten recht gut, die aus einer Simulation einer Population von 10 000 Individuen über einen Zeitraum von $t_{sim} = 10^5$ Zeitschritten gewonnen

Abbildung B.1: Vergleich Galton-Watson-Prozeß und Daten



Die Punkte stammen aus einer Simulation von 10^4 Individuen über 10^5 Zeitschritte. Die gestrichelte Linie entspricht Gleichung (B.20) und die durchgezogene Linie der Rekursion aus Gleichung (B.15) unter Verwendung von Gleichung (B.21).

wurden, wie man aus Abbildung B.1 an der gestrichelten Linie erkennen kann. Die hier vorgestellten Rechnungen haben den Vorteil, daß sie einfache analytische Formeln für die Speziesaltersverteilung liefern. Leider beschreiben sie die Daten nur für $t > 50$ (siehe Abbildung B.1) sehr gut. Man kann die Dynamik aber auch anders beschreiben, so daß die Ähnlichkeit zum Galton-Watson-Prozeß noch deutlicher wird. Man betrachtet jeden Zeitschritt als abgeschlossene Generation, so daß jedes Individuum exakt einen Zeitschritt alt wird. Dann ergibt sich sofort als erzeugende Funktion

$$f(s) = P'_0 + P'_1 s + P'_2 s^2, \quad (\text{B.21})$$

wobei die Beziehungen (B.9) gelten, die zu einem identischen Nachwuchs von $m = f'(s) = 1 - P'_0 + P'_1 M$ pro Zeitschritt führen. Nun verwendet man Gleichung (B.15), um die durchgezogene Linie in Abbildung B.1 zu erzeugen. Diese Kurve beschreibt auch für kleine t -Werte die Daten sehr gut. Aber auch diese Kurve liegt für große Speziesalter unter den gemessenen Daten. Dies läßt sich dadurch erklären, daß identische Nachkommen eines Individuums auch durch Rückmutationen entstehen können. Außerdem können mehrere Nachkommen desselben Individuums zur selben neuen Spezies gehören. Dies entspricht dem Fall, daß eine Familie nicht mit einem Mitglied, sondern mit mehreren Mitgliedern beginnt. Wie man aber aus dem Vergleich der analytischen Kurve mit den Daten schließen kann, handelt es sich hierbei nur um eine geringfügige Störung, die deshalb nicht weiter untersucht wird.

C Der Wanderer im Sumpf

C.1 Motivation

In der realen Natur führt (thermisches) Rauschen zu Mutationen und Cut-and-Splice bzw. Crossover. Im Bit-String-Modell, das eine Population wechselwirkender Individuen untersucht, wird dies in der Replikationsphase durch Anwendung der genetischen Operatoren widergespiegelt. Trotz dieser von außen vorgegebenen Veränderungen beobachtet man sehr häufig, daß die Evolution zum Stillstand kommt (siehe Kapitel 5). Im Bit-String-Modell existiert keine explizite Umwelt oder mit anderen Worten, es existiert eine statische Umwelt. Man kann sich also die Frage stellen, ob es Systeme gibt, die ohne äußere Einflüsse (quasi aus sich selbst heraus) ständig Veränderungen hervorbringen. Mit dieser Zielsetzung ist das Spielmodell des Wanderers im Sumpf untersucht worden.

In Kapitel 6 wurde ausführlich das Evolving-Bugs-Modell von PACKARD vorgestellt [4, 11, 96]. Dort wurde die Entwicklung einer Population von Individuen untersucht, die zunächst nur über die Umwelt, die aus einem quadratischen Gitter mit periodischen Randbedingungen besteht, miteinander wechselwirken. Dabei ist es schwierig, zwischen räumlichen Artefakten und Evolutionsphänomenen zu unterscheiden. Das Modell der Wanderer im Sumpf erlaubt es, die räumlichen Effekte genauer abzugrenzen, da der einzelne Wanderer *keiner* Evolution unterworfen ist.

Die in den weiteren Abschnitten dieses Anhangs dargestellten Ergebnisse sind in [41] veröffentlicht worden. Die Darstellung orientiert sich sehr stark an diesem Artikel.

C.2 Die Metapher der *Roten Königin*

Now, *here*, you see, it takes all the running *you* can do, to keep in the same place. If you want to get somewhere else, you must run at least twice as fast as that.

Diese Worte der roten Königin zu Alice stehen im Ruf, *wahrscheinlich die meistzitierten Worte aus den gesamten Alice-Büchern zu sein* [45].

Sie sind die Grundlage für das, was die Biologen manchmal als den *Red Queen Effekt* bezeichnen [89, 114]: Eine Spezies, die sich nicht verändert, zieht Parasiten und Raubtiere an. Sie erschöpft ihre Ressourcen und, falls sie selbst ein Räuber ist, entwickelt ihre Beute bessere Verteidigungsstrategien. Ein Weg, um diesen Nachteil zu umgehen, besteht darin, sich ständig zu verändern. Diese ständige Veränderung kann einfach darin bestehen, daß einzelne Individuen oder ganze Herden sich ständig bewegen. Diese Strategie ist als *kurzzeitige* Strategie für einzelne Individuen geeignet, die sich genetisch nicht anpassen können. Weit interessanter ist es aber, diese Wanderung als eine im abstrakten Raum der Eigenschaften und Strategien zu betrachten, in dem jede Spezies einen Punkt und jede Quasispezies eine Wolke von Punkten darstellt.

Natürlich wird ein solches Herumwandern durch Veränderungen in der Umwelt erzwungen, die z.B. durch klimatische Wechsel, vulkanische Aktivität, Meteoriteneinschläge, etc. ausgelöst werden. Es ist klar, daß solche von außen wirkenden Veränderungen eine starke Antriebskraft für die Evolution sind. Der dramatischste und bekannteste evolutionäre Vorfall, der wahrscheinlich durch eine solche Veränderung ausgelöst wurde, ist das Aussterben der Dinosaurier [1, 103]. Es ist aber weniger offensichtlich, ob Evolution ohne diese Störungen von außen überhaupt möglich wäre. Frühe Theorien der Reihenfolge der

Spezies (z.B. von CUVIER, siehe z.B. [97]) gehen davon aus, daß äußerliche Einflüsse die einzige treibende Kraft sind. Heute wird diese radikale Ansicht von den meisten Evolutionstheoretikern zwar nicht mehr geteilt, es herrscht aber dennoch Uneinigkeit darüber, wie wichtig innere im Vergleich zu äußeren Einflüssen sind. Während Argumente für klimatische Einflüsse in [97] gesammelt sind, sagt MAYNARD SMITH in [92]:

I see no reasons why extinctions [...] should not take place in the absence of any change in the physical environment, although [...].

Der offensichtliche Kandidat für eine solche *endogene* Evolution ist der *Red Queen Effekt*. In diesem Anhang soll keine Diskussion über seine Wichtigkeit geführt werden. Aber selbst als sekundärer Effekt sollte er einen nicht zu vernachlässigenden Einfluß haben, so daß weitergehende Untersuchungen sinnvoll sind. Es wäre von großem Interesse, wenn eine allgemeine mathematische Theorie existieren würde, die es erlauben würde, seinen Einfluß abzuschätzen. Durch das Fehlen einer solchen Theorie ist es schon interessant, zumindest ein mathematisches Modell zu haben, das explizit zeigt, daß der *Red Queen Effekt* selbst bei Abwesenheit von Zufälligkeit in der Umgebung zu Evolution führen kann.

Da kein mathematischer Begriff für *Evolution* im biologischen Sinne (d.h., Erhöhung von irgendeiner Art von Komplexität) bekannt zu sein scheint, kann hier kein solches Modell angegeben werden. In diesem Anhang soll dagegen eine Klasse von Spielmodellen vorgestellt werden, die zeigen, daß der *Red Queen Effekt*, wenn er wörtlich genommen wird, in der Lage ist, nicht stationäres und *nicht periodisches* Verhalten zu erzeugen. Die Modelle sind formal vollständig deterministisch, aber ihre Resultate sind im wesentlichen *Random-walks* (zu deutsch: Irrwege). Dies ist im Prinzip ähnlich zur Entstehung von chaotischem Verhalten in formal deterministischen, glatten, dynamischen Systemen [13], obwohl die Details doch sehr unterschiedlich sind. Im nächsten Abschnitt wird die einfachste Version vorgestellt. Variationen werden in Abschnitt C.4 beschrieben und in Abschnitt C.5 wird das Ganze noch einmal diskutiert.

Zuvor sollen noch kurz die sonst üblichen Modelle erwähnt werden: Üblicherweise werden Räuber-Beute-Beziehungen durch LOTKA-VOLTERRA-Gleichungen beschrieben. Es ist gezeigt worden [60], daß LOTKA-VOLTERRA-Systeme mit mehr als zwei Partnern in der Lage sind, chaotische Bewegungen zu erzeugen. Dies ist zwar noch keine wahre evolutionäre Aktivität, aber es ist das Beste, was man von Differentialgleichungssystemen mit wenigen Freiheitsgraden erwarten kann. Es scheint, daß chaotisches Verhalten eine notwendige Voraussetzung für Evolution ist. Die in diesem Anhang vorgestellten Modelle unterscheiden sich sehr stark von der Phasenraumstruktur der *Lotka-Volterra*-Modelle, obwohl sie versuchen, einen im wesentlichen ähnlichen Mechanismus zu beschreiben. Sie könnten a priori zu wirklicher Evolution führen, was die ursprüngliche Motivation für ihre Untersuchung war.

C.3 Ein einzelner Wanderer im Sumpf

C.3.1 Beschreibung des Modells

In der einfachsten Version des Modells wird ein einzelner Wanderer auf einem endlichen, zweidimensionalen und quadratischen Gitter der Größe $L \times L$ betrachtet. Es gelten periodische Randbedingungen, so daß eine Torusgeometrie entsteht. Jeder Gitterplatz i des Gitters besitzt eine *Einsinktiefe* h_i . Zu Beginn (d.h. zur Zeit $t = 0$) ist die Landschaft flach, so daß $h_i = 0$ für alle Gitterplätze gilt. Der Wanderer startet an einem zufälligen Platz (z.B. am Ursprung) und schaut in eine zufällige Richtung.

Die Tiefe h_i mißt, in welchem Maße der Wanderer in der Vergangenheit Eindrücke auf dem Gitterplatz i hinterlassen hat. Während der Wanderer sich auf der Stelle i befindet, sinkt er ein, als stünde er im Sumpf. Die Tiefe des Bodens steigt um einen festen Wert pro Zeiteinheit. Dies sei eine Einheit der Tiefe:

$$h_i^{t+1} = h_i^t + 1. \quad (\text{C.1})$$

Ein nicht durch einen Wanderer besetzter Platz j relaxiert pro Zeitschritt nach dem Gesetz

$$h_j^{t+1} = f(h_j^t). \quad (\text{C.2})$$

Die detaillierte Form der Funktion $f(h)$ ist für die Dynamik unwesentlich. Es ist nur wichtig, daß

$$f(h) \geq 0, \quad f(h) < h \quad \text{für} \quad h > 0, \quad df/dh \geq 0, \quad f(0) = 0 \quad (\text{C.3})$$

und

$$f(h) < 1 \quad \text{für} \quad 0 < h \leq 2 \quad (\text{C.4})$$

gilt. Gleichung (C.3) bedeutet, daß ein Gitterplatz, der nicht vom Wanderer besucht wird, monoton gegen $h = 0$ relaxiert. In Abschnitt C.4 wird diskutiert, was passiert, wenn Gleichung (C.4) fallengelassen wird.

Die Dynamik des Modells wird dadurch festgelegt, daß der Wanderer immer auf den Nachbarplatz geht, der die geringste Tiefe hat. Auf diese Weise versucht er, die Eindrücke im Boden, die er selber erzeugt hat, zu umgehen. Dies wird in einer kurzfristigen Weise ohne langfristige Strategie gemacht. Die Entscheidung wird nur anhand der Tiefen des eigenen Platzes und der vier Nachbarplätze getroffen.

C.3.2 Erste Konsequenzen

Die Gleichungen (C.3) und (C.4) beinhalten, daß der Wanderer niemals still steht oder sich um 180 Grad dreht. Der Wanderer (d.h. die *rote Königin*) läuft also immer, so wie es sich Lewis Carroll vorgestellt hat. Damit das Stehenbleiben ein optimaler Schritt wäre, müßte

$$1 + h_i < f(h_j) \quad (\text{C.5})$$

für irgendein Paar $\{h_i, h_j\}$ gelten. Dies ist nicht möglich. In gleicher Weise sind Drehungen um 180 Grad (das sind *Rückschritte*) nicht möglich, da sie

$$f(1 + h_i) < f(f(h_j)) \quad (\text{C.6})$$

benötigen, was ebenfalls nicht sein kann.

Wenn zwei oder mehr Gitterplätze mit derselben niedrigsten Tiefe existieren, verwendet der Wanderer eine deterministische Regel, die bestimmt, wohin er geht. Wenn nur isotrope (rotationssymmetrische) Regeln verwendet werden, existieren 3 mögliche Regeln und ihre spiegelsymmetrischen Partner. Sie werden definiert durch die Reihenfolge der bevorzugten Bewegungsrichtungen: (1) geradeaus, links, rechts (Regel GLR); (2) links, geradeaus, rechts (LGR); und (3) links, rechts, geradeaus (LRG). Wie weiter unten gezeigt wird, beeinflußt die gewählte Regel zwar das Verhalten auf kleinen Gittern und für kurze Zeiten, aber sie ist irrelevant für das Langzeitverhalten auf großen Gittern.

Aus den obigen Definitionen ergeben sich eine Anzahl einfacher Theoreme.

Zunächst einmal gilt, daß der Wanderer bei einem unendlich langen Weg das gesamte Gitter bedeckt, d.h., es kann keinen Gitterplatz geben, der niemals besucht wird. Es gilt sogar, daß jeder Gitterplatz unendlich oft besucht wird. Für den Beweis dieser Behauptung bedenke man, daß das Gitter endlich ist. Damit müssen *einige* Plätze unendlich oft besucht werden. Man betrachte nun einen solchen Gitterplatz und nehme an, daß er einen Nachbarn j hat, der nur endlich oft besucht wird. Nach einer endlichen Zeit bevorzugt der Wanderer die Richtungen, die nicht zu j führen. Dies gilt, obwohl die Tiefe h_j gegen Null relaxiert und damit irgendwann niedriger als die übrigen Tiefen wird. Dies kann offensichtlich nicht sein.

Dieses vorige Argument läßt sich noch verschärfen. Wegen der Gleichungen (C.3) und (C.4) ist die Reihenfolge der Tiefen der Nachbarn von Gitterplatz i genau die Reihenfolge der Zeitabstände seit dem letzten Besuch dieser Gitterplätze. Dies bestimmt auch die Reihenfolge, in der die Nachbarplätze wieder besucht werden, wenn der Wanderer den Platz i das nächste Mal verläßt, solange der Nachbar in der Zwischenzeit nicht in einem Weg enthalten gewesen ist, der *nicht* durch den Gitterplatz i geführt hat. Dies bedeutet, daß die kontinuierliche Variable h_i durch eine diskrete *spin*artige Variable ersetzt werden kann, die die Reihenfolge der vier Nachbarn speichert. Eine solche Variable hat $4! = 24$ mögliche Werte, $s_i = 1, \dots, 24$. (Ähnliche Spin-Modelle sind von COHEN *et al.* [79, 106] und BUNIMOVICH *et al.* [18] untersucht worden.) Bei Verwendung dieser Variablen wird die Dynamik allerdings recht kompliziert. Es ändert sich nämlich nicht nur die Spineinstellung des besuchten Gitterplatzes, sondern auch die aller seiner vier Nachbarn. Das wichtigste Ergebnis aber ist, daß die Spindynamik (und damit auch die Wege) unabhängig von der detaillierten Form der Funktion $f(h)$ ist.

Für dieses Modell erhält man einen endlichen Phasenraum, der $N(L) = 24^{L^2}$ mögliche Zustände enthält, so daß ein Weg des Wanderers irgendwann periodisch werden muß.

C.3.3 Erste Simulationsergebnisse

Für kleine Gitter kann man die Periodendauern und auch die periodischen Orbits selbst relativ leicht finden. Sie hängen, wie schon erwähnt, von der speziellen Implementation ab, also von den bevorzugten Richtungen bei gleich hohen Nachbarplätzen. Die Periodendauern hängen auch sehr stark von der Gittergröße L ab. Im allgemeinen findet man kein monotonen Verhalten. In Tabelle C.1 sind Abschätzungen für die Transientenzeiten für kleine Gitter für die drei oben genannten Regeln zusammengefaßt.

Man erhält diese Abschätzungen, indem man gleichzeitig 2 Gitter simuliert, die beide mit einer flachen Landschaft beginnen, von denen aber das eine mit doppelter Geschwindigkeit simuliert wird. Die Zeiten in Tabelle C.1 sind die Zeiten, bei denen die Tiefenverteilungen zum ersten Mal übereinstimmen. Man kann sich leicht überzeugen, daß diese Zeiten von unten durch die Transientenzeit T_0 und von oben durch $T_0 + T_p$ begrenzt sind. Dabei ist T_p die Periodendauer des periodischen Orbits. Für die Regel GLR steigt T_0 nur quadratisch mit der Gittergröße L an, wenn $L \neq 4n + 1$ gilt. Ansonsten scheint T_0 ungefähr wie e^{aL^2} anzusteigen. Die Konstante a ist dabei deutlich kleiner als $\ln 24$, was andeutet, daß ein Großteil des Phasenraumes für einen Weg, der auf einer flachen Oberfläche startet, nicht zugänglich ist.

Zum Abschluß dieser Diskussion soll noch bemerkt werden, daß die periodischen Orbits im allgemeinen nicht alle Gitterplätze gleich häufig besuchen, wenn sie auch jeden Platz mindestens einmal pro Orbit enthalten. Damit führt der Weg nicht zu einer Gleichverteilung.

Tabelle C.1: Transientenzeiten bis zum Erreichen eines periodischen Orbits

L	Regel		
	GLR	LRG	LGR
2	8	8	8
3	18	18	18
4	48	48	48
5	286	150	75
6	72	108	144
7	98	816	1 666
8	192	256	2 432
9	3 608	1 411	5 146
10	200	26 580	18 100
11	242	5 166	11 193
12	432	54 458	456 624
13	1 413 720	338 275	2 503 976
14	392	2 622 708	207 600
15	450	5 289 144	1 072 802
16	768	41 426 880	48 245 600
17	335 203 407	245 621 434	103 122 840
\vdots	\vdots	\vdots	\vdots
$4n$	$3L^2$	–	–
$4n + 1$	–	–	–
$4n + 2$	$2L^2$	–	–
$4n + 3$	$2L^2$	–	–

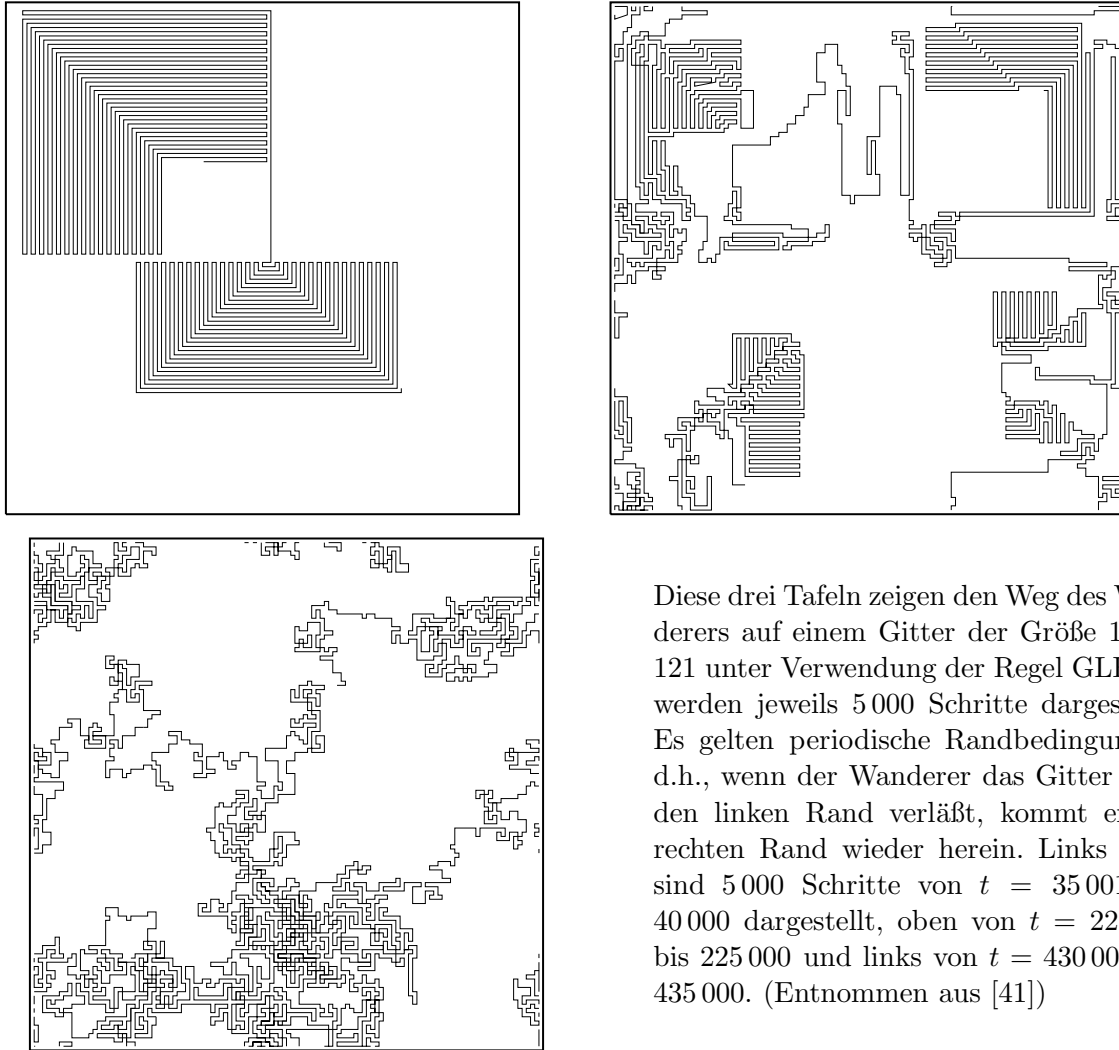
Diese Tabelle zeigt Abschätzungen für die Transientenzeiten T_0 bis zum Erreichen eines periodischen Orbits für verschiedene Gittergrößen und für die drei Regeln GLR, LGR und LRG. Die Regeln bezeichnen die Reihenfolge der Bevorzugung der drei Richtungen Geradeaus (G), Links (L) und Rechts (R).

Man erhält diese Abschätzungen, indem man gleichzeitig 2 Gitter simuliert, die beide mit einer flachen Landschaft beginnen, von denen aber das eine mit doppelter Geschwindigkeit simuliert wird. Die Zeiten geben an, wann die Tiefenverteilungen zum ersten Mal übereinstimmen. Diese Zeiten werden von unten durch die Transientenzeit T_0 und von oben durch $T_0 + T_p$ begrenzt, wobei T_p die Periode des periodischen Orbits ist. (Entnommen aus [41])

Für große Werte von L (und für $L = 4n + 1$ im Falle von Regel GLR) sind die Transientenzeiten so lang, daß nur noch das transiente Verhalten an sich von praktischer Bedeutung ist: Der Weg eines Wanderers beginnt sehr regulär und wird mit steigender Zeit t mehr und mehr *zufällig*. Dies wird in den drei Tafeln von Abbildung C.1 verdeutlicht. Jede Tafel zeigt 5 000 Schritte eines Weges auf einem Gitter der Größe 121×121 für die Regel GLR. Nach etwa $N = 2 \times 10^5$ Schritten sind noch sehr regelmäßige Wegstrecken sichtbar. Der Weg wird aber immer unregelmäßiger und nach $N = 4 \times 10^5$ sieht der Weg vollständig zufällig aus. In Abbildung C.2 werden 20 000 Schritte eines Weges gezeigt, der etwa 10^7 Schritte von der flachen Landschaft des Anfangs entfernt ist. Vergleicht man Gitter bis zur Größe $L \simeq 1 400$, so findet man eine typische Zeit $T_r \approx L^{2.5}$, die der Wanderer benötigt, damit der Weg für alle drei Regeln zufällig aussieht.

Nach dieser Zeit T_r zeigt der Weg dasselbe statistische Verhalten für alle drei Regeln. Dies wird in Abbildung C.3 deutlich, wo die Anzahl der Geradeaus Schritte (Rhomben), Linksdrehungen (Quadrate) und Rechtsdrehungen (Pluszeichen), die während eines 10^6 Schritte langen Weges stattgefunden haben, als Funktion der Zeit dargestellt werden. Die Gittergröße beträgt $L = 1001$. Das Verhalten hängt für kleine Zeiten von der gewählten Regel ab, aber für große Zeiten $N > 10^8$ ist es für alle drei Regeln gleich. Das statistische Verhalten ist auch dann gleich, wenn nicht auf einer flachen Landschaft, sondern auf einer Zufallslandschaft gestartet wird, auf der die Tiefen h_i auf zufällige Werte zwischen 0 und 1 gesetzt werden. In diesem Fall produzieren alle drei Regeln identische Wege. Man kann daher vermuten, daß der Prozeß ergodisch und unabhängig von der gewählten Regel im Grenzfall unendlicher Gitter wird. Dies wird auch durch die weiteren Resultate gestützt.

Abbildung C.1: Die Wege des Wanderers



Diese drei Tafeln zeigen den Weg des Wanderers auf einem Gitter der Größe 121×121 unter Verwendung der Regel GLR. Es werden jeweils 5 000 Schritte dargestellt. Es gelten periodische Randbedingungen, d.h., wenn der Wanderer das Gitter über den linken Rand verläßt, kommt er am rechten Rand wieder herein. Links oben sind 5 000 Schritte von $t = 35\,001$ bis $40\,000$ dargestellt, oben von $t = 220\,001$ bis $225\,000$ und links von $t = 430\,001$ bis $435\,000$. (Entnommen aus [41])

Ein sehr bemerkenswertes Resultat, das sich aus Abbildung C.3 ergibt, besagt, daß die Zahlen der Links- und Rechtsdrehungen bis auf geringe Fluktuationen identisch sind. Damit wird die chirale Symmetrie, die von den einzelnen Regeln explizit gebrochen wird, statistisch wiederhergestellt.

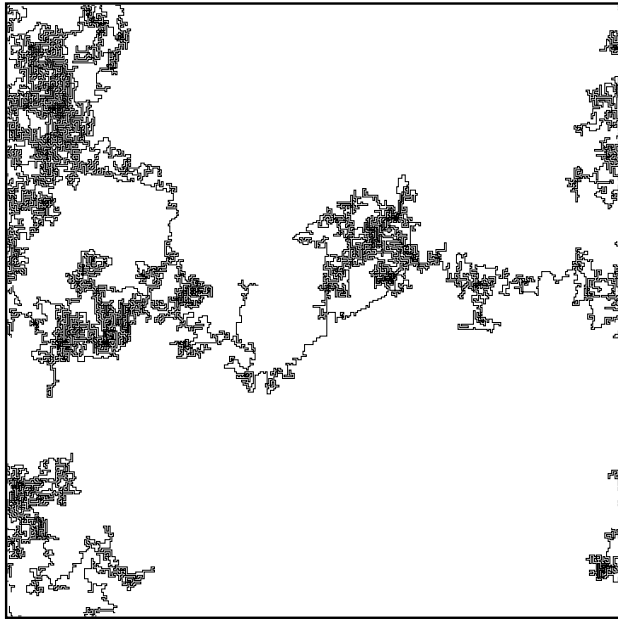
C.3.4 Die statistischen Eigenschaften des Modells

Im Folgenden werden die statistischen Eigenschaften von Wegen untersucht, die aus der geordneten Anfangsphase heraus sind (d.h. $N \gg T_r$). In Abbildung C.4 wird die Verteilung $P(l)$ der Wegabschnitte dargestellt, die mindestens l Schritte hintereinander in dieselbe Richtung gehen. Mit Ausnahme der ersten beiden Punkte liegen die Werte auf einer perfekten Exponentialfunktion. Dies bedeutet, daß der Wanderer ein Kurzzeitgedächtnis besitzt, was wiederum nahelegt, daß asymptotisch ein echtes Randomwalk-Verhalten vorliegt. Für dieses gilt:

$$R_n^2 \equiv \langle |\mathbf{x}_{t+n} - \mathbf{x}_t|^2 \rangle \sim n. \quad (\text{C.7})$$

Um dieses direkt zu testen, wird in Abbildung C.5 das normierte mittlere Abstandsquadrat R_n^2/n gegen den zeitlichen Abstand n dargestellt. In Abbildung C.6 wird die

Abbildung C.2: Der Weg des Wanderers



Diese Abbildung zeigt ein Gitter der Größe 361×361 . Nachdem der Weg innerhalb von etwa 10^7 Schritten immer zufälliger geworden ist, werden 20 000 Schritte eines Wanderers gezeigt, der die Regel GLR verwendet. Es gelten ebenfalls wieder periodische Randbedingungen. (Entnommen aus [41])

Autokorrelationsfunktion

$$c_n = \langle \Delta \mathbf{x}_{t+n} \cdot \Delta \mathbf{x}_t \rangle \quad (\text{C.8})$$

dargestellt, dabei gilt: $\Delta \mathbf{x}_t = \mathbf{x}_{t+1} - \mathbf{x}_t$. Beide Abbildungen zeigen in etwa, aber nicht genau, das erwartete Verhalten von Randomwalks. Für $n \leq 200$ kann man aus der Autokorrelation eher ein Potenzgesetz mit dem Exponenten $\alpha \approx -0.9$ ablesen. Diesem sind starke periodische Oszillationen überlagert, außerdem erkennt man deutliche Abweichungen bei kleinem n . Dies würde nahelegen, daß $R_n^2 \sim n^{2-\alpha} = n^{1.1}$ gilt. Aus Abbildung C.5 erkennt man, daß dies tatsächlich das Verhalten für $n \approx 100$ ist. Für große n neigt sich die Kurve aber weiter herunter, so daß asymptotisch doch $R_n^2 \sim n$ gilt. Versucht man logarithmische Korrekturen der Form

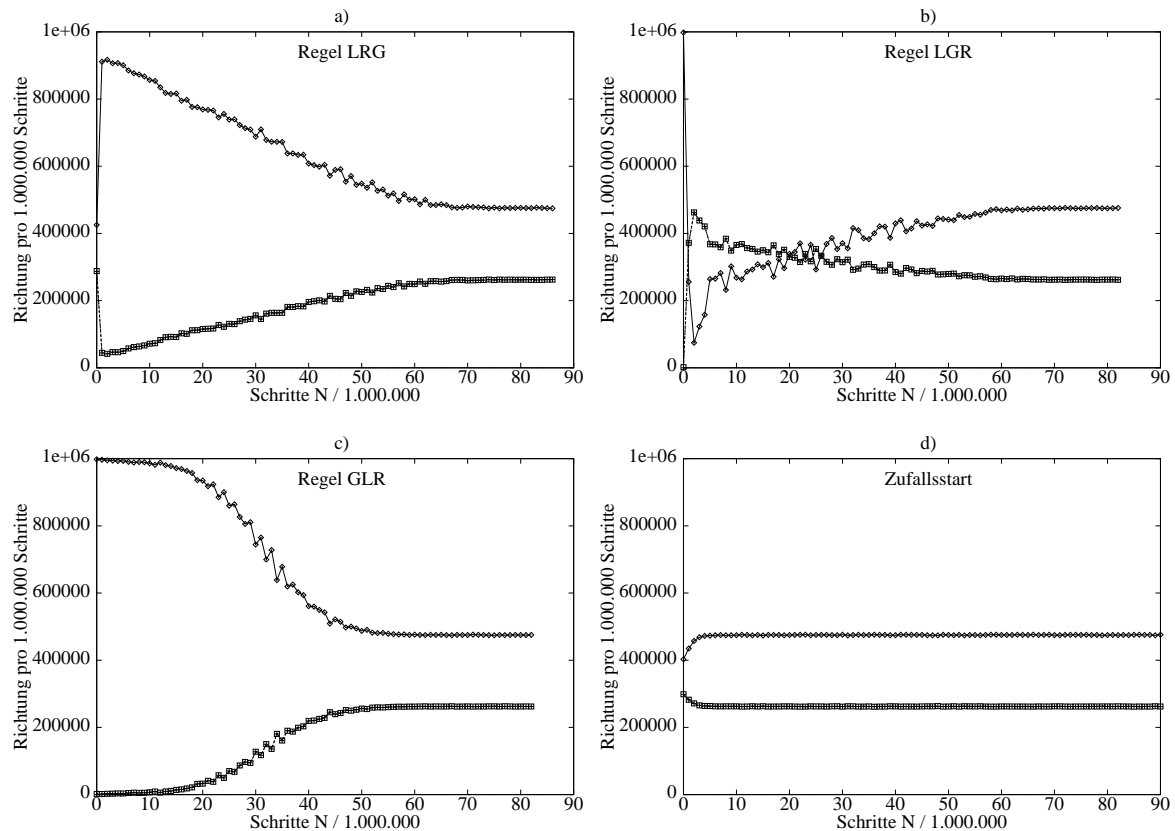
$$R_n^2 \sim n(\ln n)^\beta, \quad (\text{C.9})$$

dann findet man $\beta \approx 0.33$ mit einem Fehler von etwa ± 0.05 wegen der deutlichen Abweichungen von Gleichung (C.9). Man sieht also, daß der Weg asymptotisch ein (deterministischer!) Randomwalk ist, für den eventuell logarithmische Korrekturen gelten. Dies deutet an, daß man sich exakt an der oberen kritischen Dimension dieses Problems befindet, d.h., die obere kritische Dimension ist $d = 2$.

Die letzte Beobachtung zeigt im wesentlichen, daß die Tendenz, die eigenen Spuren zu vermeiden, ausreicht, um die chirale Symmetrie wiederherzustellen. Sie reicht aber nicht aus, um nicht triviales Langzeitverhalten zu bekommen. Dies steht im Unterschied zu *self avoiding walks* (SAW, zu deutsch: selbstvermeidende Irrwege) [26], bei denen der Wanderer vernichtet wird, wenn er in seine eigene Spur laufen will. Das Verhalten ist aber ähnlich zu dem von sogenannten *true self avoiding walks* [2] (TSAW, die auch als *self-repelling walks* bezeichnet werden [19]). Bei diesen versucht der Wanderer seiner eigenen Spur auszuweichen, wird aber nicht vernichtet, wenn dies nicht möglich ist.

Bei beiden (SAW und TSAW) handelt es sich um Zufallsprozesse. Für SAW gilt im Gegensatz zu diesem Modell $R_n^2 \sim n^{2\nu}$ mit $\nu > 1/2$. Für TSAW gilt in 2 Dimensionen bis auf logarithmische Korrekturen ebenfalls $R_n^2 \sim n$. Die Selbstvermeidung ist in diesem Modell nicht sehr stark, was man auch an der Verteilung der Wiederkehrzeiten erkennen kann.

Abbildung C.3: Die regelabhängige Einschwingphase



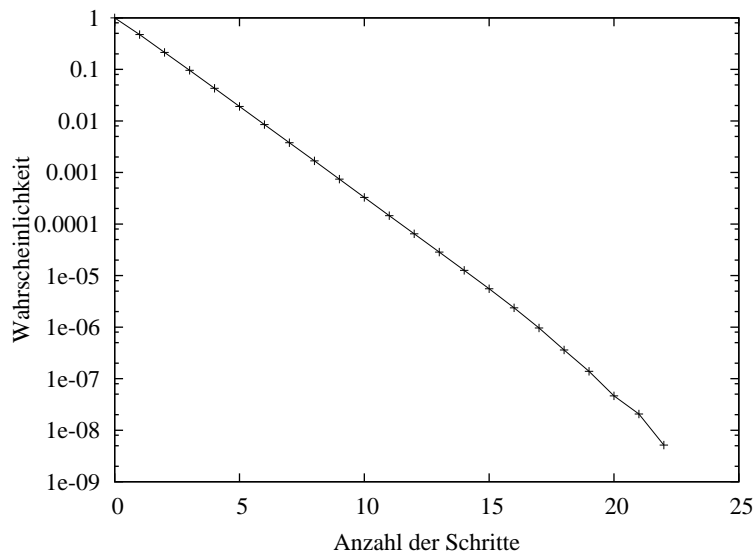
Die Anzahl der Geradeaus Schritte (Rhomben, die oberen Kurven in den Tafeln), Linksdrehungen (Quadrate, eine der beiden unteren Kurven) und Rechtsdrehungen (Pluszeichen, die zweite untere Kurve) die während 10^6 Schritten auf einem Gitter der Größe $L = 1001$ stattgefunden haben. Die einzelnen Tafeln entsprechen: Regel LRG (a); Regel LGR (b); Regel GLR (c); zufälligem Start (d). Innerhalb der Genauigkeit des Diagramms liegen die Quadrate und Pluszeichen aufeinander. (Entnommen aus [41])

Die Wiederkehrzeit t ist definiert als die Zahl von Schritten, die der Wanderer zwischen zwei aufeinanderfolgenden Besuchen desselben Gitterplatzes geht. Für zweidimensionale Randomwalks ist bekannt [110], daß bis auf logarithmische Korrekturen $P(t) \sim 1/t$ gilt. Für SAW ist $P(t) \equiv 0$. In Abbildung C.7 ist $P(t)$ für den Wanderer dargestellt. Bis auf Abweichungen bei kleinem t sieht man in der Tat einen Abfall $\sim 1/t$. Damit reicht die Tendenz, seine eigene Spur zu vermeiden, aus, den Weg zufällig zu machen. Sie genügt aber nicht, das Kreuzen der eigenen Spur zu verhindern.

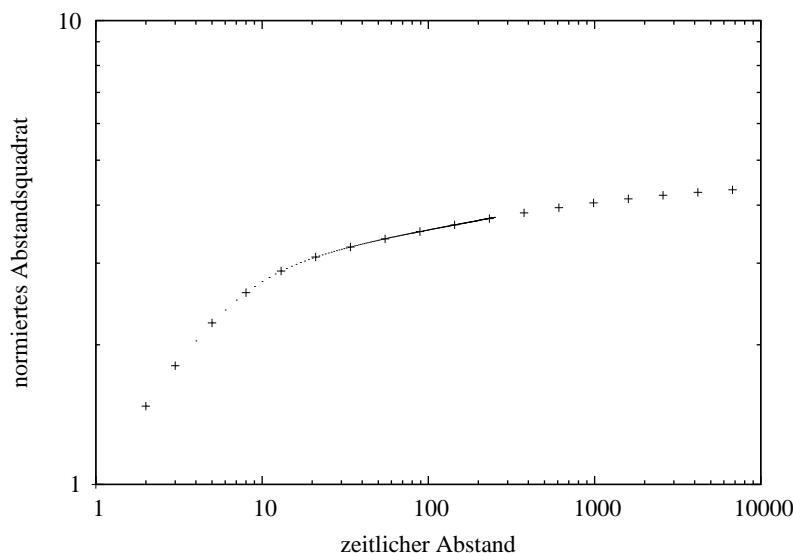
C.4 Erweiterungen und Verallgemeinerungen

C.4.1 Dynamischer Zufall

In der einfachen Version des Modells hat sich der Wanderer vollständig deterministisch verhalten. Man kann aber genauso gut probabilistische Versionen der obigen Regeln betrachten, in denen bei jedem Schritt die Regel mit der Wahrscheinlichkeit $1 - p$ befolgt wird, während mit Wahrscheinlichkeit p ein zufälliger Schritt stattfindet. Das Ergebnis ist so, wie man es erwarten würde: Für alle drei Regeln werden die Wege bei großen Gittern und bei Zeiten die wesentlich größer als L^2 sind zufällig. Darüberhinaus verändern sich die statistischen Eigenschaften kontinuierlich für kleine Werte von p und gehen in die Eigenschaften für den Grenzfall $p \rightarrow 0$ über. Bei der Regel GLR gilt dies natürlich

Abbildung C.4: Die Verteilung $P(l)$ 

Diese Abbildung zeigt für den statistischen Gleichgewichtszustand die Wahrscheinlichkeit, mindestens l Schritte in dieselbe Richtung zu gehen. Für diese und die weiteren Abbildungen wurden Gitter mit $L > 1000$ simuliert. Die ersten $T = 200 \times L^2$ Schritte wurden verworfen und die nächsten M Schritte ($M > 10^8$) wurden zur Bestimmung der Mittelwerte verwendet. (Entnommen aus [41])

Abbildung C.5: Das mittlere Abstandsquadrat $\langle R_n^2 \rangle / n$ 

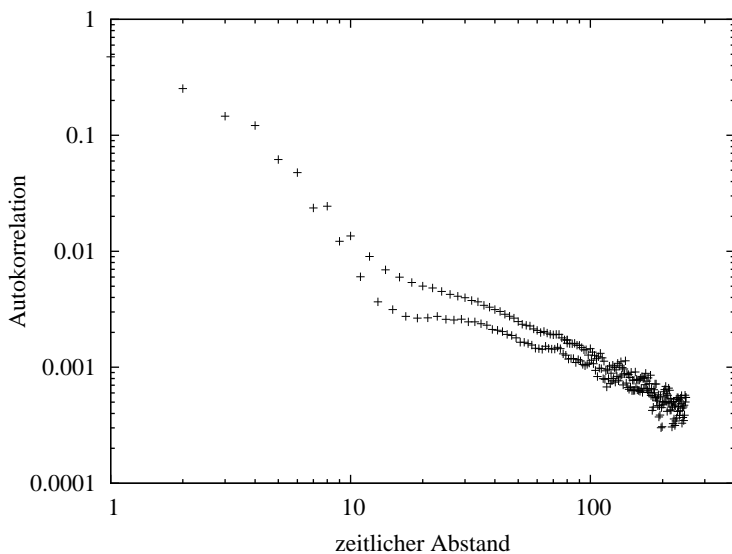
Doppeltlogarithmische Auftragung des mittleren Abstandsquadrates gegen den zeitlichen Abstand n . Um die Proportionalität deutlicher zu machen, ist auf der y -Achse $\langle R_n^2 \rangle / n$ aufgetragen. Für einen Randomwalk erwartet man asymptotische Konvergenz gegen eine horizontale Linie. Punkte und Rhomben stammen von zwei verschiedenen Gittergrößen. (Entnommen aus [41])

nur für Gitter der Größe $L = 4n + 1$. Dies zeigt, daß das nicht zufällige Verhalten bei quadratischen Gittern der übrigen Größen untypisch und instabil ist.

Die Stabilität des pseudozufälligen Verhaltens gegen wirkliches, zufälliges Rauschen wird ebenfalls in den Varianten der weiteren Abschnitte beobachtet.

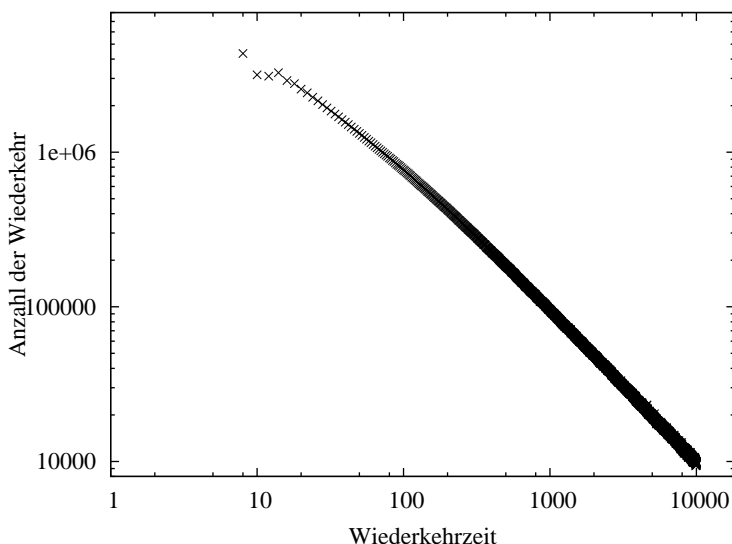
C.4.2 Unendliche Gitter

Bisher ist nur der Fall eines endlichen Gitters mit flachen Anfangsbedingungen untersucht worden. Für ein unendliches und ursprünglich flaches Gitter ist das Problem natürlich trivial, denn der Wanderer läuft ohne Richtungswechsel in die einmal eingeschlagene Richtung. Man kann aber auch ein unendliches Gitter mit einer *nicht flachen* Anfangsbedingung betrachten. Würde der Wanderer keine Eindrücke hinterlassen, dann wäre sein Weg zufällig und damit in 2 Dimensionen wiederkehrend. Die Tatsache, daß die Spuren, die er hinterläßt, ebenfalls einen Randomwalk erzeugen, deutet an, daß man ebenfalls $R_n^2 \sim n$ erhält.

Abbildung C.6: Die Autokorrelation c_n 

Diese Abbildung zeigt in doppeltlogarithmischer Darstellung die Autokorrelation c_n , die in Gleichung C.8 definiert worden ist. (Entnommen aus [41])

Abbildung C.7: Die Verteilung der Wiederkehrzeiten



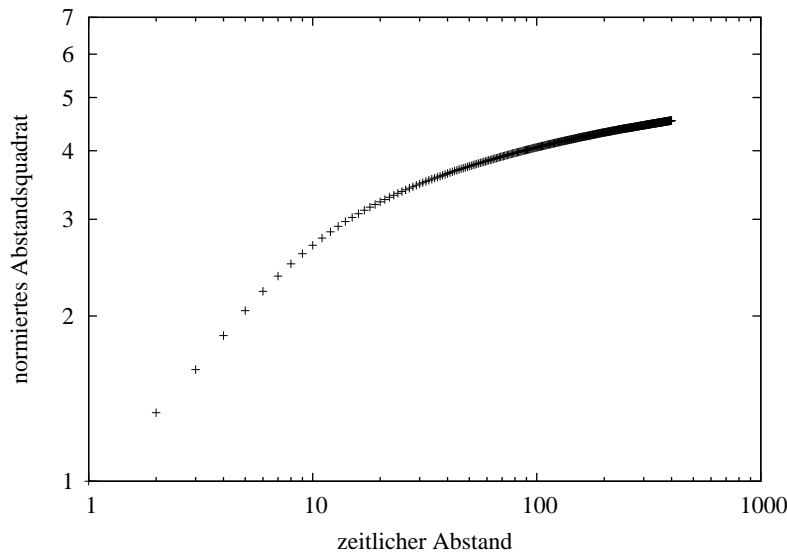
Diese Abbildung zeigt in doppeltlogarithmischer Darstellung die Wiederkehrzeiten für einen Wanderer auf einem 961×961 Gitter bei einem Weg, der aus 2×10^9 Schritten besteht. (Entnommen aus [41])

Dies wird in der Tat in Simulationen beobachtet. In Abbildung C.8 ist R_n^2/n für 3×10^6 Wege dargestellt, die jeder auf einer neuen Zufallslandschaft starten. Man sieht, daß die Wege noch etwas *aufgeblasener* als in Abbildung C.5 sind. Man erkennt auch wieder logarithmische Korrekturen $R_n^2 \sim n(\ln n)^\beta$, diesmal aber mit $\beta \approx 0.4$ (wieder mit großer Ungenauigkeit).

C.4.3 Langsame Relaxation

Es wurde schon mehrfach darauf hingewiesen, daß die bisherigen Resultate nicht von der speziellen Form der Funktion $f(h)$ abhängen, solange Gleichung (C.4) erfüllt ist. Sie garantiert, daß das Verhalten des Wanderers von den Zeiten bestimmt wird, die zwischen den Besuchen eines Gitterplatzes vergangen sind. Es findet also keine Ansammlung der Tiefe über mehrere Besuche statt. Dies muß nicht mehr gelten, wenn die Relaxation genügend langsam abläuft.

Man kann beobachten, daß Gleichung (C.4) durch wesentlich schwächere Bedingungen

Abbildung C.8: Das mittlere Abstandsquadrat auf unendlichen Gittern

Diese Abbildung zeigt in doppeltlogarithmischer Darstellung den mittleren, quadratischen End-zu-End Abstand $\langle R_n^2 \rangle / n$ für kurze Wege auf einer zufälligen Landschaft. Das Gitter ist genügend groß, so daß ein Überschreiten der Ränder nicht vorkommt. (Entnommen aus [41])

ersetzt werden kann, ohne daß sich die Ergebnisse verändern. Es ist allerdings schwierig, die schwächsten Bedingungen dafür zu finden, daß die Wege identisch mit denen aus den vorigen Abschnitten bleiben. Aus diesem Grund wird an dieser Stelle nicht weiter darauf eingegangen.

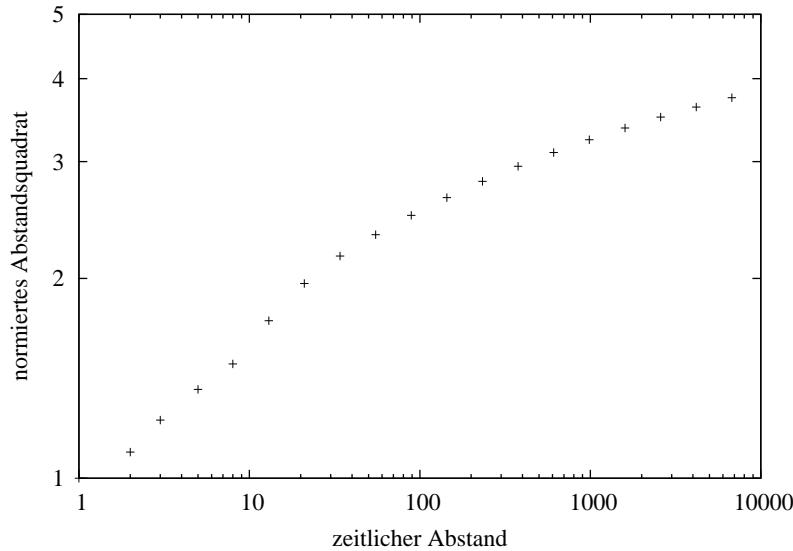
Wenn die Relaxation noch genügend schnell ist, unterscheidet sich der Weg gelegentlich von einem solchen mit schneller Relaxation, die Langzeitstatistik und das qualitative Verhalten bleiben allerdings gleich. Dies gilt nicht mehr, wenn die Relaxation sehr langsam wird, oder wenn überhaupt keine Relaxation mehr stattfindet. Im Rest dieses Abschnitts wird der Grenzfall ohne Relaxation untersucht.

In diesem Fall wird die Entscheidung des Wanderers nicht mehr von der zeitlichen Reihenfolge der Besuche bestimmt, sondern nur noch von der Anzahl der Besuche. Der Wanderer geht zu dem Nachbarplatz, den er bisher am seltensten besucht hat. Damit sind auch Drehungen um 180 Grad und sogar das Stillstehen mögliche Alternativen, solange sie nicht explizit verboten sind. Die chirale Symmetrie wird nun nicht mehr wiederhergestellt.

Im Folgenden sei angenommen, daß Rückschritte und Stehenbleiben verboten sind. Dies verhindert, daß sich die Zahl der möglichen Regeln zu stark vermehrt. Bei den Regeln GLR und LGR wird der Weg auf allen quadratischen Gittern mit flachen Anfangsbedingungen geordnet und periodisch. Er wird mit sehr langen und langsamen Transienten ungeordnet auf Gittern, die geringe Unordnung, wie z.B. einem Gitterplatz mit $h_i \neq 0$, enthalten.

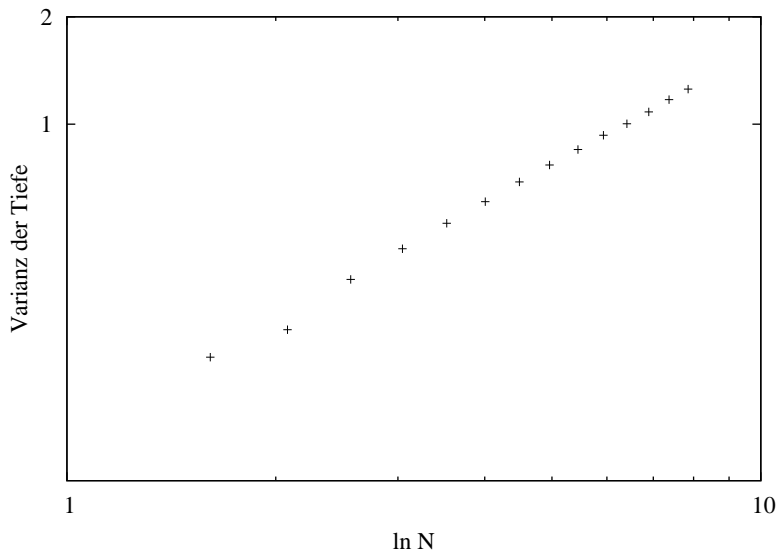
Im Gegensatz dazu wird der Weg sehr schnell zufällig bei Verwendung von Regel LRG, auch dann schon, wenn die Oberfläche zu Anfang flach ist. Die Tiefenverteilung wird dabei zu einer zufälligen Landschaft.

In Abbildung C.9 wird $\langle R_n^2 \rangle / n$ im stationären Zustand gezeigt. Man sieht wieder, daß der End-zu-End Abstand etwas größer als beim reinen Randomwalk ist. Es gilt aber bis auf logarithmische Korrekturen wieder asymptotisch $\sim \sqrt{n}$ ($\beta = 0.61$). Hieraus könnte man naiv schließen, daß die Landschaft im Langzeit-Limes zu einer Zufalls Oberfläche wird. Dies ist aber definitiv nicht der Fall, wie man aus Abbildung C.10 ablesen kann. Dort ist die Varianz der Tiefenverteilung in Abhängigkeit von der Gittergröße L dargestellt. Um die Varianz zu berechnen, sind die Mittelwerte nur zu solchen Zeiten durchgeführt worden, die ganzzahlige Vielfache der Gittergröße L^2 sind. Dies vermeidet *halbgefüllte Ebenen*. Man sieht, daß die Varianz der Tiefenverteilung nur etwa wie $(\log L)^{1.1}$ statt wie $\sim L$ ansteigt.

Abbildung C.9: Die Abstandsverteilung ohne Relaxation

Diese Abbildung zeigt in doppeltlogarithmischer Darstellung den mittleren, normierten End-zu-End Abstand $\langle R_n^2 \rangle / n$ gegen die Zahl der Schritte n . Es wird Regel LRG verwendet und es findet keine Relaxation der Landschaft statt, d.h., $f(h) = h$. (Entnommen aus [41])

Man erhält qualitativ ähnliche Resultate, wenn Rückwärtsschritte erlaubt werden.

Abbildung C.10: Die Varianz der Tiefenverteilung

Diese Abbildung zeigt in doppeltlogarithmischer Darstellung die Varianz der Tiefenverteilung gegen den *Logarithmus* der Gittergröße L . Die Werte von L sind die Fibonaccizahlen $5, 8, \dots, 2548$. Wie schon in Abbildung C.9 wird Regel LRG verwendet und es findet ebenfalls keine Relaxation statt. (Entnommen aus [41])

Zum Schluß sei noch darauf hingewiesen, daß das Modell ohne Relaxation im wesentlichen einer deterministische Version des *true self-avoiding walk* Problems entspricht [2]. Auch dafür wird vermutet, daß $\langle R_n^2 \rangle \sim n$ und $\delta h \approx \text{const}$ bis auf logarithmische Korrekturen gilt [98].

C.4.4 Hügelige Landschaften

Bei den bisherigen Varianten des Modells ist die Landschaft immer gegen eine flache Landschaft $h_i = 0$ relaxiert. Stattdessen soll nun der Fall betrachtet werden, bei der jeder Gitterplatz i eine andere Funktion $f_i(h)$ mit einem Fixpunkt h_i^* hat. Genauer gesagt, wird

$$f_i(h) = f(h - h_i^*) + h_i^* \quad (\text{C.10})$$

mit demselben $f(h)$ wie bisher gewählt. Die Relaxation führt nun zu einer hügeligen Landschaft $h_i = h_i^*$. Der Wanderer versucht immer noch, den höchsten Punkt in seiner

Umgebung zu erreichen.

Dieses Modell ist biologisch vermutlich interessanter als das Modell aus Abschnitt C.3, da es den *Red Queen Effekt* in einer Situation beschreibt, in der nicht alle Zustände a priori gleich gut sind. Wenn der Wanderer keine Spuren hinterläßt, so ergibt sich nur ein einfaches Optimierungsproblem, bei dem der Wanderer das nächstliegende Maximum aufsucht und dort bleibt. Die *Weichheit* des Bodens verhindert das, besonders dann, wenn die Relaxation langsam ist.

Die Spitzen der Hügel entsprechen ökologischen Nischen, die von einer unbeweglichen Spezies nur für einen endlichen Zeitraum besetzt werden können. Diese können später erneut besucht werden, wenn sie wieder gute Nischen sind. Wenn eine Nische breit und tief genug ist, wird der Wanderer niemals vollständig von ihr weggetrieben werden. Er ist vielmehr in der Lage, in einem Teil der Nische zu verweilen, bis der Rest der Nische wieder relaxiert ist. Damit ist, wenn die Landschaft flacher und/oder das Optimum schmaler wird, ein Übergang zwischen lokalisierten und delokalisierten Wegen zu erwarten. Im delokalisierten Bereich sind typische Wege intermittent, da der Wanderer ein lokales Maximum anstrebt, in seiner Umgebung verweilt, bis seine Spuren das Maximum zerstört haben, und dann zu einem anderen lokalen Maximum wechselt. Sowohl im lokalisierten, als auch im delokalisierten Bereich existieren *keine* periodischen Orbits, es sei denn, die Nische ist sehr schmal *und* sehr tief.

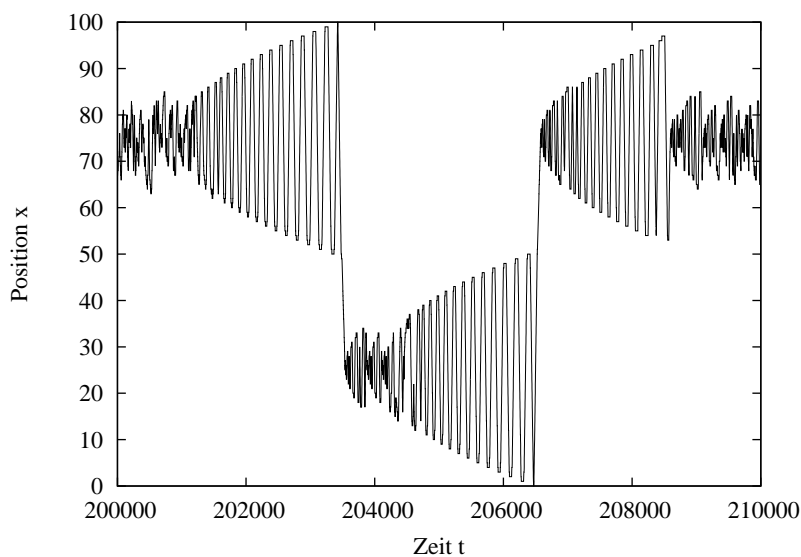
Die Details der Dynamik hängen nicht nur von den Eigenschaften der Oberfläche $\{h_i^*\}$, sondern auch von der Geschwindigkeit der Relaxation, also von $f(h)$, ab. In Abbildung C.11 ist die x -Koordinate eines Wanderers in einer Landschaft mit zwei Tälern der Form

$$h^*(x, y) = a \cdot \sin(2\pi x/L) \cdot \sin(2\pi y/L) \quad (\text{C.11})$$

dargestellt. Dabei gilt $a = 0.05$, $L = 100$ und

$$f(h) = \frac{h}{1+h}. \quad (\text{C.12})$$

Abbildung C.11: Der Weg in einer hügeligen Landschaft



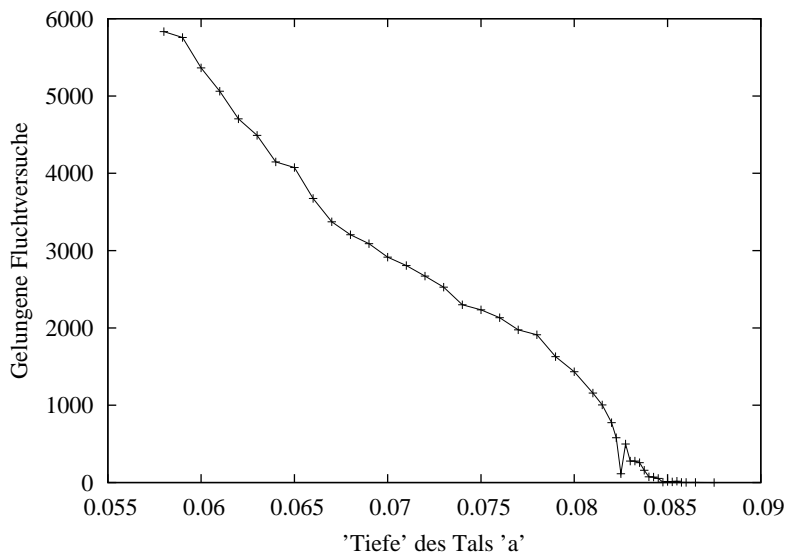
Diese Abbildung zeigt das intermittente Verhalten eines Wanderers in einer Landschaft mit zwei Tälern, die durch die Gleichung (C.11) mit $a = 0.05$ und $L = 100$ gegeben ist. Zur Relaxation wurde Gleichung (C.12) verwendet. (Entnommen aus [41])

Man sieht in der Tat die erwartete Intermittenz. Die Schwelle zur Delokalisation liegt bei $a \simeq 0.085$. Bei einer Oberfläche mit Nischen (bzw. Hügeln) unterschiedlicher Höhe kann

dies zu anomaler Diffusion [17] führen, die durch eine breite Verteilung von *Fluchtzeiten* entsteht.

Als letztes ist noch untersucht worden, ob der Übergang von der Lokalisation zur Delokalisation scharf ist. Wenn das Entkommen aus der Nische dem thermisch aktivierten Entweichen aus einem Potentialtopf entspricht, dann ist die Rate des Entkommens niemals exakt Null, da sie exponentiell mit der Tiefe des Topfes absinkt. Andererseits müßte der Übergang scharf sein, wenn das Verhalten ähnlich dem einer *Crisis* [53] in einem glatten, dynamischen System wäre. In Abbildung C.12 ist die Fluchtrate gegen die *Tiefe* a des Maximums dargestellt. Es ist wieder eine Landschaft mit zwei Tälern nach Gleichung (C.11) mit $L = 100$ verwendet worden. Selbst durch sehr lange Läufe (5×10^8 Schritte) konnte nicht geklärt werden, welche der beiden Möglichkeiten realisiert ist. Wenn der Übergang scharf sein sollte, dann steigt die Fluchtrate jedenfalls sehr langsam mit einer Potenz $(a - a_{cr})^\gamma$ mit $\gamma > 1$ an.

Abbildung C.12: Flucht aus einem Maximum



Diese Abbildung zeigt mit welcher Rate ein Wanderer das Maximum verläßt, in dem er sich gerade befindet. Es ist die Landschaft, die durch Gleichung (C.11) mit $L = 100$ beschrieben wird, verwendet worden. Es wird die Zahl der Schritte, die aus dem Maximum herausgeführt haben, bei einem Weg, der aus 10^8 Schritten besteht, gegen die *Tiefe* a des Maximums dargestellt.

C.4.5 Mehrere Wanderer

Als letzte Variante werden mehrere Wanderer untersucht, die gleichzeitig auf einer großen, aber endlichen und zu Beginn flachen Landschaft laufen.

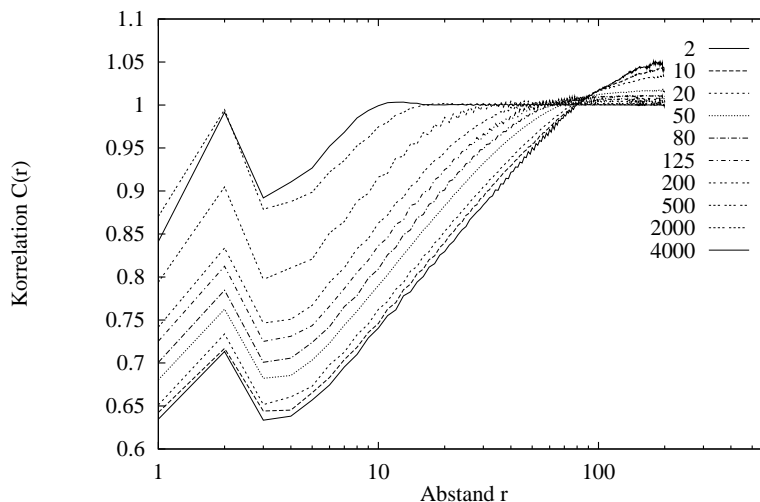
Es existieren zwei mögliche Vorgehensweisen. Bei der Ersten entscheiden alle Wanderer simultan in welche Richtung sie gehen, und führen dann – ebenfalls simultan – diesen Schritt aus. Dies kann man relativ leicht auf einer massiv parallelen Maschine wie der *Connection Machine* durchführen. Es führt im Laufe der Simulation zu einer *Kondensation* aller Wanderer. Sobald mindestens zwei Wanderer auf demselben Gitterplatz stehen, wählen sie in Zukunft immer dieselbe Richtung und bleiben daher zusammen. Gruppen von Wanderern können dann aus verschiedenen Richtungen denselben Gitterplatz wählen, so daß immer größere Gruppen entstehen.

Die zweite Möglichkeit besteht darin, daß alle Wanderer nacheinander vorgehen. Jeder entscheidet, wohin er geht, und führt diesen Schritt aus, bevor der Nächste an die Reihe kommt. Auf diese Weise kann der nächste Wanderer schon den Eindruck sehen, den der erste hinterläßt. Dies kann leider nicht mehr sinnvoll auf einer *Connection Machine* implementiert werden. Es führt zum genauen Gegenteil der ersten Möglichkeit, denn

selbst wenn alle Wanderer auf derselben Stelle starten, so laufen sie nun auseinander. Alle Resultate, die im Folgenden vorgestellt werden, wurden mit der zweiten Möglichkeit erhalten.

Da die Wanderer die Eindrücke vermeiden, die von vorherigen Schritten erzeugt wurden, fühlen sie effektiv eine abstoßende Kraft, die zu negativen Korrelationen führt. Obwohl dieser Effekt bei der Selbstabstoßung sehr gering ist, wie man aus Abbildung C.5 ablesen kann, ist die abstoßende Korrelation sehr stark. Dies erkennt man in Abbildung C.13, die die normalisierte 2-Punkt-Dichtekorrelation $C(r)$ als Funktion des *Manhattan*-Abstandes für 10 verschiedene Wandererdichten zeigt. Sowohl die Stärke, als auch die Reichweite der Korrelation sinkt mit dem Ansteigen der Zahl N der Wanderer. Sie ist also am stärksten für $N = 2$. Man könnte vermuten, daß die Reichweite der Korrelation durch den mittleren Abstand der nächsten Nachbarn gegeben ist, dies ist aber offensichtlich nicht der Fall. Wie Abbildung C.13 zeigt ist der Anstieg mit N viel schwächer als $1/\sqrt{N}$ und er kann für $L = 201$ nur für $N > 10$ deutlich gesehen werden. Mit Ausnahme sehr hoher Dichten existiert eine Region von r , in der $C(r)$ linear in $\log r$ ist und in der die Steigung dichteunabhängig ist.

Abbildung C.13: Die Abstandskorrelation für mehrere Wanderer

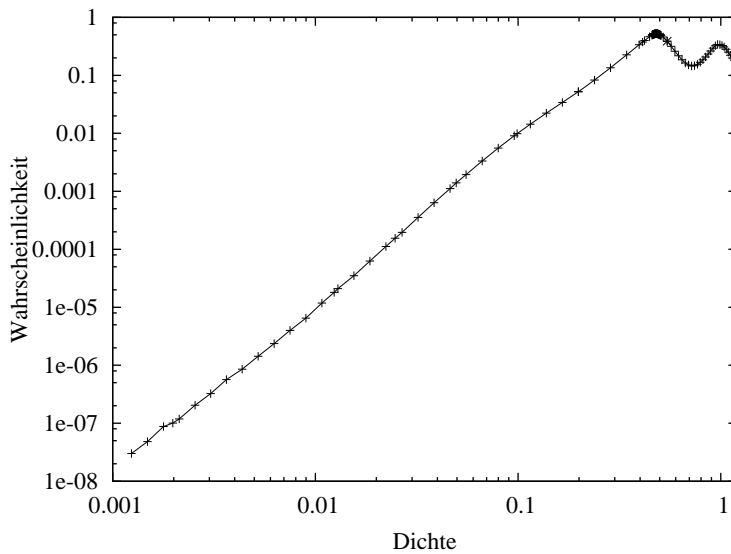


Diese Abbildung zeigt Abstandskorrelation $C(r)$ zwischen Wanderern auf einem Gitter der Größe $L = 201$ gegen den *Manhattan*-Abstand $r = |\delta x| + |\delta y|$. Die Normalisierung ist so gewählt, daß die Korrelation 1 für unkorrelierte Wanderer beträgt. (Entnommen aus [41])

Ein interessanter Effekt ist das lokale Maximum der Korrelation bei dem *Manhattan*-Abstand 2, das in Abbildung C.13 deutlich sichtbar ist. Dieses Maximum wird verursacht von Wandererpaaren, für die $\delta x = \pm 1$ **und** $\delta y = \pm 1$ gilt. Betrachtet man die Entwicklung auf dem Bildschirm, so sieht man, daß diese diagonal liegenden Paare eine Tendenz haben, für mehrere Zeitschritte zusammenzubleiben. Es ist zu vermuten, daß dieser Effekt ein Artefakt der gewählten Nachbarschaft ist.

In Abschnitt C.3.3 wurde festgestellt, daß die chirale Symmetrie statistisch wiederhergestellt wurde, obwohl sie explizit von jeder Bewegungsregel gebrochen wurde. Wenn mehrere Wanderer vorhanden sind, muß dies nicht mehr gelten. Aber auch in diesem Fall wird die chirale Symmetrie statistisch wiederhergestellt. Ein weiteres interessantes Ergebnis wird in Abbildung C.14 dargestellt. Ein einzelner Wanderer führt auch dann, wenn es erlaubt ist, niemals Rückwärtsschritte (d.h. 180 Grad Drehungen) aus. Bei mehreren Wanderern kann dies aber der günstigste Schritt sein, wenn all drei anderen Nachbarn schon besetzt sind. Damit sollte in erster Näherung die Wahrscheinlichkeit für Rückschritte mit der dritten Potenz der Wandererdichte ρ ansteigen. Für hohe Dichten muß die Kurve ab-

Abbildung C.14: Wahrscheinlichkeit für Rückschritte



Diese Abbildung zeigt die Wahrscheinlichkeit für Rückschritte in Abhängigkeit von der Dichte der Wanderer ρ . Die Gittergröße beträgt jeweils $L = 201$. Es wurde bei gleicher Höhe die Regel GRHL verwendet, H steht dabei für hinten. Man beobachtet ein Ansteigen der Rückschritte mit ρ^3 . Für hohe Dichten erkennt man ein Abflachen, das vorhanden sein muß, da die Wahrscheinlichkeit nicht über 1 ansteigen kann. Der genaue Verlauf der Kurve ist allerdings bisher nicht vollständig verstanden.

flachen, da die Wahrscheinlichkeit nicht über 1 ansteigen kann. Dies wird durch Abbildung C.14 bestätigt, ein erstes Abweichen findet schon bei $\rho \simeq 0.1$ statt. Bei $\rho \simeq 0.5$ durchläuft die Wahrscheinlichkeit ein Maximum. Der weitere Verlauf hat entfernte Ähnlichkeit mit einer Sinuskurve. Dieser Verlauf der Kurve ist allerdings bisher noch nicht verstanden.

C.5 Diskussion des Modells

Die Wege, die in diesem Anhang untersucht worden sind, haben Ähnlichkeiten mit Randomwalks mit Gedächtnis, die in den letzten Jahren in der Physik untersucht worden sind. Beispiele hierfür sind *self-avoiding walks* (SAW) [26], *'true' self-avoiding walks* (TSAW) [2, 19] und *laplacian walks* [88] (ein Überblick wird in [98] gegeben). Im Gegensatz zu dem Modell der Wanderer sind die anderen Modelle stochastische Prozesse, d.h., die Wege sind wirklich *zufällige* Wege. In diesem Anhang war es eines der Hauptinteressen, herauszufinden, ob eine schwache Form der Selbstvermeidung zu zufälligem oder vielleicht sogar zu strukturell komplexem Verhalten in einem vollständig deterministischen System führen kann. Dies ist tatsächlich der Fall. Mehrere Varianten des Modells führen zu Wegen, deren End-zu-End Abstand asymptotisch wie \sqrt{n} skaliert. In all diesen Fällen sind starke Korrekturen vorhanden, die mit logarithmischen Korrekturen vereinbar sind. Dies deutet an, daß $d = 2$ die obere kritische Dimension ist. Damit sind die hier untersuchten Wege ähnlich dem TSAW-Problem [2, 19].

Das Modell der Wanderer ist auch den Modellen mit umklappenden Spiegeln aus [18, 79, 106] ähnlich. Diese Modelle sind wie das vorliegende deterministisch, sie finden aber in einer Landschaft statt, die ungeordnete Anfangsbedingungen hat. Die Zufälligkeit der Trajektorien dieser Modelle ist eine direkte Folge der Zufälligkeit in den Anfangsbedingungen. In diesem Modell dagegen kann man selbst bei mehreren Wanderern mit einem hochgeordneten Zustand, bei dem alle Wanderer auf demselben Gitterplatz starten, anfangen und nach einer kurzen *Einschwingphase* ergeben sich dieselben Ergebnisse, die schon in Abschnitt C.4.5 beschrieben wurden. Darüberhinaus führt die Veränderung der Landschaft in den Spiegelmodellen nicht zu einer Selbstabstoßung.

Während bisher die Wege als Metaphern für das *Umherwandern* im abstrakten Raum der phänotypischen Eigenschaften angesehen wurde, kann man sie genauso (vielleicht sogar

realistischer) als wirkliche Wege eines Futter suchenden Tieres ansehen. Damit zeigen die Ergebnisse dieses Anhangs, daß die kurzsichtige Suche nach den am wenigsten *abgegrasten* Stellen in einer sonst gleichförmigen Umwelt dafür sorgt, daß ein zufälliges Herumirren auch dann entsteht, wenn eine deterministische Strategie verwendet wird. Dies muß nicht als Nachteil der kurzsichtigen Suche angesehen werden. Das Tier könnte sich auch weiter umsehen und den optimalen Ort in einer größeren Umgebung suchen, damit würde es aber Gefahr laufen, statt des vielen kurzreichweitigen Umherirrens weniger, aber größere Desaster zu erleiden. In [64] wird ein ähnliches Beispiel für Computernetzwerke diskutiert. Dort findet man chaotisches Verhalten, das mit ausgefeilteren Strategien immer stärker wird.

Wie schon in Abschnitt C.2 bemerkt wurde, ist die Tatsache, daß ein System, das **nicht** von außen gestört wird, dennoch aperiodisches und unvorhersagbares Verhalten zeigen kann, nicht neu. Dieses sollte ganz offensichtlich auch für biologische Systeme gelten. Es scheint aber, daß die Folgerungen, die man daraus ziehen kann, nicht immer richtig gewürdigt werden. Der hier besprochene Modellansatz scheint zwar neu zu sein, aber sowohl die Möglichkeit von spontanem chaotischen Verhalten als auch seine Wichtigkeit für die Evolution sind schon verschiedentlich herausgehoben worden. Die bestuntersuchten Modelle, die zu chaotischem Verhalten führen können, sind n -Spezies Lotka-Volterra Systeme mit $n \geq 3$ [60] oder 2-Spezies Lotka-Volterra Systeme mit räumlichen Freiheitsgraden [121]. Von Bedeutung in diesem Zusammenhang sind auch die Ergebnisse von LINDGREN [85], der im *Iterated Prisoner's Dilemma* (siehe auch Kapitel 2) immer ausgefeiltere (und daher ständig wechselnde) Strategien findet, oder auch die Ergebnisse des Bit-String-Modells, die in Kapitel 5 beschrieben wurden. In Verbindung mit der Notwendigkeit von chaotischem Verhalten in Ökologien sollte auch das Konzept des *Homeochaos*, das in [67] eingeführt wurde, erwähnt werden.

Als die Arbeit am Wanderermodell begonnen wurde, bestand die Hoffnung, daß nicht nur zufälliges, sondern auch *komplexes* Verhalten gefunden würde. Damit ist Verhalten mit *struktureller* Komplexität, im Gegensatz zu *algorithmischer* Komplexität, wie sie von KOLMOGOROV [78], CHAITIN [22] und anderen definiert wird, gemeint. In Systemen, in denen nur ein bestimmtes, mittleres Verhalten von Interesse ist, ist algorithmische Komplexität kein gutes Maß, da sie nicht zwischen Zufälligkeit und struktureller Komplexität unterscheiden kann [52]. Die bestuntersuchte Klasse von Modellen, die zu solchem Verhalten führen kann, sind *zellulare Automaten* [118]. Obwohl das Wanderermodell als zellulärer Automat mit sehr wenigen Aktionen angesehen werden kann, ist kein wirklich komplexes Verhalten gefunden worden. Dies kann allerdings daran liegen, daß es sehr subtil versteckt ist [50] und daher bisher übersehen wurde. Außerdem können die logarithmischen Effekte andeuten, daß man sich am Rande von komplexem Verhalten befindet.

Als letztes soll noch bemerkt werden, daß der Grenzfall unendlich langsamer Relaxation zu einer neuen Klasse von Modellen für das Oberflächenwachstum führt. Bisherige Modelle [115] beinhalten gleichmäßiges Wachstum an allen Oberflächenplätzen (Eden growth), bevorzugtes Wachstum an offenen Spitzen (DLA=Diffusion Limited Aggregation) und Wachstum an zeitlich aktivierten Stellen, die ihre Aktivität von ihren Nachbarn erben (Epidemiemodelle). All diese Prozesse führen zu Höhenfluktuationen der Oberfläche, die mit einer Potenz der Oberfläche anwachsen. Im Gegensatz dazu wachsen die Höhenfluktuationen im Wanderermodell nur logarithmisch an. Ob dies zu Anwendungen in der Festkörperphysik führen kann, ist bisher noch unbekannt.

Literaturverzeichnis

- [1] W. Alvarez, F. Asaro, H.V. Michel, and L.W. Alvarez, *Iridium Anomaly Approximately Synchronous with Terminal Eocene Extinctions*, *Science* **216**, pp 886-888, 1982.
- [2] D. Amit, G. Parisi and L. Peliti, *The Asymptotic Behavior of the 'True' Self-Avoiding Walk*, *Phys. Rev.* **B27**, pp 1635-1645, 1983.
- [3] P.W. Anderson, *Suggested model for prebiotic evolution: The use of chaos*, *Proc. Natl. Acad. Sci. USA* **80**, pp 3386-3390, 1983.
- [4] A.M. Assad and N.H. Packard, *Emergent Colonization in an Artificial Ecology*, in *Proceedings of the 1st European Conference on Artificial Life*, ECAL 91, MIT Press, 1992.
- [5] R. Axelrod, *The Evolution of Cooperation*, New York: Basic Books, 1984.
- [6] R.J. Bagley, *The Functional Self-Organization of Autocatalytic Networks in a Model of the Evolution of Biogenesis*, PhD Thesis, University of California at San Diego, La Jolla, CA, 1991.
- [7] R.J. Bagley, J.D. Farmer, S.A. Kauffman, N.H. Packard, A.S. Perelson, and I.M. Stadnyk, *Modeling Adaptive Biological Systems*, *BioSystems*, **23**, pp 113-138, 1989.
- [8] R.J. Bagley and J.D. Farmer, *Spontaneous Emergence of a Metabolism*, in [84], pp 93-140.
- [9] R.J. Bagley, J.D. Farmer and W. Fontana, *Evolution of a Metabolism*, in [84], pp 141-158.
- [10] P. Bak, C. Tang, and K. Wiesenfeld, *Self-Organized Criticality*, *Phys. Rev.* **A38**, pp 364-374, 1988.
- [11] M.A. Bedau and N.H. Packard, *Measurement of Evolutionary Activity, Teleology and Life*, in [84], pp 431-461.
- [12] R.K. Belew, *Evolution, Learning, and Culture: Computational Metaphors for Adaptive Algorithms*, *Complex Systems* **4**, 11-49, 1990.
- [13] P. Bergé, Y. Pomeau and Ch. Vidal, *Order within Chaos*, Wiley, 1986.
- [14] H.T. Betteridge, *Cassell's German-English, English-German Dictionary*, Macmillan Publishing Company, New York, 1978.
- [15] M.C. Boerlijst and P. Hogeweg, *Spiral Waves Structure in Pre-Biotic Evolution: Hypercycles stable against Parasites*, *Physica* **D48**, pp 17-28, 1991.
- [16] M. Boerlijst and P. Hogeweg, *Self-Structuring and Selection: Spiral Waves as a Substrate for Prebiotic Evolution*, in [84], pp 255-276.
- [17] J.-P. Bouchaud and A. Georges, *Anomalous diffusion in disordered media*, *Physics Reports* **195**, pp 127-293, 1990.
- [18] L. Bunimovich and S.E. Troubetzkoy, *Recurrence Properties of Lorentz Lattice Gas Cellular Automata*, *J. Stat. Phys.* **67**, pp 289-302, 1992.
- [19] C. Byrnes and A.J. Guttmann, *On self-repelling walks*, *J. Phys.* **A17**, pp 3335-3342, 1984.
- [20] P. Cariani, *Emergence and Artificial Life*, in [84], pp 775-798.
- [21] H.A. Ceccatto and B.A. Huberman, *The Complexity of Hierarchical Systems*, *Physica Scripta* **37**, 145-150, 1988.

- [22] G.J. Chaitin, *Towards a mathematical definition of 'life'*, in *The Maximum Entropy Principle*, R.D. Levine and M. Tribus, eds., MIT Press, Cambridge, MA, 1979.
- [23] A. Church, *The Calculi of Lambda-Conversion*, Annals of Mathematics Studies no. 6, Princeton University Press, Princeton, 1941.
- [24] Paul Churchland, *Reduction, Qualia, and the direct Introspection of Brain States*, Journal of Philosophy **82**, pp 8-28, 1985.
- [25] E.C. Cox and T.C. Gibson, *Selection for high mutation rates in chemostats*, Genetics **77**, pp 169-184, 1974.
- [26] P.G. deGennes, *Scaling Concepts in Polymer Physics*, Cornell University Press, Ithaca, 1979.
- [27] B. Derrida and L. Peliti, *Evolution in a Flat Fitness Landscape*, Bulletin of Mathematical Biology **53**, pp 355-382, 1991.
- [28] T.O. Diener, *Circular RNAs: Relics of precellular evolution?*, Proc. Natl. Acad. Sci. USA **86**, pp 9370-9374, 1989.
- [29] M. Eigen, *Stufen zum Leben*, Piper, München, 1987.
- [30] M. Eigen and P. Schuster, *The Hypercycle*, Springer-Verlag, Berlin, 1979.
- [31] N. Elredge and S.J. Gould, *Punctuated Equilibria: An Alternative to Phyletic Gradualism*, in *Models in Paleobiology*, Ed. T.J.M Schopf, Freeman Cooper, Berlin, pp 82-115, 1972.
- [32] A.K. Erlang, *Problem 15*, Matem. Tidsskr. **B**, p 36, 1929.
- [33] J.D. Farmer, S.A. Kauffman, and N.H. Packard, *Autocatalytic Replication of Polymers*, Physica **D22**, pp 50-67, 1986.
- [34] R.A. Fisher, *The distribution of gene ratios for rare mutations*, Proc. Royal Soc. Edinburgh **50**, pp 204-219, 1930.
- [35] R.A. Fisher *The genetical theory of natural selection*, Oxford, 1930.
- [36] E.T. Freeman, M.W. Lugowsky, *Artificial Life Digest*, `alife-request` at `iuvax.cs.indiana.edu`, siehe vor allem Ausgaben #46, #49, and #50 (1989-1991).
- [37] W. Fontana, *Algorithmic Chemistry: A model for functional self-organization*. Technical Report LA-UR-90-1959, Los Alamos National Laboratory, 1990;
W. Fontana, *Algorithmic Chemistry*, in [84], pp 159-209.
- [38] W. Fontana, W. Schnabl, and P. Schuster, *Physical aspects of evolutionary optimization and adaption*, Phys. Rev. **A40**, 3301-3321, 1989.
- [39] H. Freund and P. Grassberger, *Multispin coding for spin glasses*, J. Phys. A **21**, L801-L805, 1988.
- [40] Harald Freund and Peter Grassberger, *The ground state of the $\pm J$ spin-glass from a heuristic matching algorithm*, J. Phys. A **22**, 4045-4059, 1989.
- [41] Harald Freund and Peter Grassberger, *The Red Queen's Walk*, Wuppertal Preprint WUB 92-22, erscheint in Physica A, 1992.
- [42] H. Freund and R. Wolter, *Evolution of Bitstrings: Some Preliminary Results*, Complex Systems **5**, pp 279-298, 1991.
- [43] H. Freund and R. Wolter, in Vorbereitung.
- [44] F. Galton, *Problem 4001*, Educational Times, p 17, 1 April, 1873.

- [45] M. Gardner / L. Carroll, *The Annotated Alice*, p.210, Clarkson N. Potter, Inc., New York, 1960.
- [46] M.R. Garey and D.S. Johnson, *Computers and Intractability : A Guide to NP-Completeness*, Freeman, San Francisco, 1979.
- [47] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
- [48] D.E. Goldberg, B. Korb, and K. Deb, *Messy Genetic Algorithms: Motivation, Analysis, and First Results*, *Complex Systems* **3**, pp 493-539, 1989.
- [49] D.E. Goldberg, K. Deb, and B. Korb, *Messy Genetic Algorithms Revisited: Studies in Mixed Size and Scale*, *Complex Systems* **4**, pp 415-444, 1990.
- [50] P. Grassberger, *Long-Range Effects in an Elementary Cellular Automaton*, *J. Stat. Phys.* **45**, pp 27-39, 1986.
- [51] P. Grassberger, *Toward a Quantitative Theory of Self-Generated Complexity*, *Int. J. Theo. Phys.* **25**, pp 907-938, 1986.
- [52] P. Grassberger, *Randomness, Information, and Complexity*, in *Proc. of 5th Mexican Summer School on Stat. Phys.*, F. Ramos-Gomez, ed., World Scientific, Singapore, 1991.
- [53] C. Grebogi, E. Ott and J.A. Yorke, *Crises, Sudden Changes in Chaotic Attractors and Transient Chaos*, *Physica* **D 7**, pp 181-200, 1983.
- [54] J.B.S. Haldane, *A mathematical theory of natural and artificial selection, V*, *Proc. Cambridge Phil. Soc.* **23**, pp 838-844, 1927.
- [55] T.E. Harris, *The Theory of Branching Processes*, Springer-Verlag, Berlin, 1963.
- [56] P.G. Higgs and B. Derrida, *Stochastic models for species formation in evolving populations*, *J. Phys.* **A24**, L985-L991, 1991.
- [57] P.G. Higgs and B. Derrida, *Genetic distance and species formation in evolving populations*, Saclay Preprint, Saclay SPhT/91-184, 1991, Submitted to *J. Mol. Evol.*
- [58] W.D. Hillis, *Co-Evolving Parasites Improve Simulated Evolution as an Optimization Procedure*, in [84], pp 313-324.
- [59] G.E. Hinton and S.J. Nowlan, *How Learning Can Guide Evolution*, *Complex Systems* **1**, 495-502, 1987.
- [60] J. Hofbauer and K. Sigmund, *The Theory of Evolution and Dynamical Systems*, Cambridge University Press, Cambridge, 1988.
- [61] J.H. Holland, *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, 1975.
- [62] J.E. Hopcroft and J.D. Ullman, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, Reading, 1979.
- [63] B.A. Huberman and T. Hogg, *Complexity and Adaptation*, *Physica* **D22**, 376-384, 1986.
- [64] B.A. Huberman and T. Hogg, in *Ecology of Computation*, p. 77, North Holland, Amsterdam, 1988.
- [65] T. Ikegami and K. Kaneko, *Computer Symbiosis – Emergence of Symbiotic Behavior through Evolution*, *Physica* **D42**, pp 235-243, 1990.

- [66] D. Jefferson, R. Collins, C. Cooper, M. Dyer, M. Flowers, R. Korf, C. Taylor, and A. Wang, *Evolution as a Theme in Artificial Life: The Genesys/Tracker System*, in [84], pp 549-578.
- [67] K. Kaneko and T. Ikegami, *Homeochaos: dynamic stability of a symbiotic network with population dynamics and evolving mutation rates*, *Physica D* **56**, pp 406-429, 1992.
- [68] S.A. Kauffman, *Autocatalytic Sets of Proteins*, *Journal of Theoretical Biology* **119**, pp 1-24, 1986.
- [69] S.A. Kauffman, *Adaption on Rugged Fitness Landscapes*, In *Lectures in the Science of Complexity*, edited by D.L. Stein, Santa Fe Institute Studies in the Science of Complexity, Lec. Vol. I, pp 527-618, Redwood City, CA: Addison Wesley, 1989.
- [70] S.A. Kauffman, *Requirements for Evolvability in Complex System: Orderly Dynamics and Frozen Components*, pp 151-192 in *Complexity, Entropy and the Physics of Information* edited by W.H. Zurek, Addison Wesley Publishing Company, Redwood City, 1990.
- [71] S.A. Kauffman and S. Johnson, *Co-Evolution to the Edge of Chaos: Coupled Fitness Landscapes, Poised States, and Co-Evolutionary Avalanches*, in [84], pp 325-370.
- [72] M. Kimura, *The Neutral Theory of Molecular Evolution*, *Scientific American* **241**, November 1979, pp 94-104, 1979.
- [73] H.P. Kleber und D. Schlee, *Biochemie*, (Teil 1: Allgemeine und funktionelle Biochemie), Gustav Fischer Verlag, Jena, 1991.
- [74] S. Kirkpatrick and E.P. Stoll, *A Very Fast Shift-Register Sequence Random Number Generator*, *Journal of Computational Physics*, **40**, pp 517-526, 1981.
- [75] R. Knippers, P. Philippsen, K.P. Schäfer und E. Fanning, *Molekulare Genetik*, Georg Thieme Verlag, Stuttgart, 1990.
- [76] D. Knuth, *The Art of Computer Programming, Volume 3: Sorting and Searching*, Addison-Wesley, Reading, MA, 1973.
- [77] A. Kolmogorov and N.A. Dmitriev, *Branching stochastic processes*, *Doklady Akad. Nauk U.S.S.R.* **56**, pp 5-8, 1947.
- [78] A.N. Kolmogorov, *Three approaches to the quantitative definition of information*, *Problems of Inform. Theory* **1**, 3-7, 1965.
- [79] X.P. Kong and E.G.D. Cohen, *Diffusion and Propagation in Triangular Lorentz Lattice Gas Cellular Automata*, *J. Stat. Phys.* **62**, pp 737-757, 1991.
- [80] A. Kornberg, *DNA Synthesis*, W.H. Freeman, San Francisco, 1974.
- [81] J.R. Koza, *Genetic Evolution and Co-Evolution of Computer Programs*, in [84], pp 603-629.
- [82] C.G. Langton (Ed.), *Artificial Life*, Addison Wesley Publishing Company, Redwood City, 1989.
- [83] C.G. Langton, *Artificial Life*, in [82], pp 1-47.
- [84] C.G. Langton, C. Taylor, J.D. Farmer and S. Rasmussen (Eds.), *Artificial Life II*, Addison Wesley Publishing Company, Redwood City, 1992.
- [85] K. Lindgren, *Evolutionary Phenomena in Simple Dynamics*, in [84], pp 295-312.
- [86] A.J. Lotka, *The extinction of families, I*, *J. Washington Acad. Sci.* **21**, pp 377-380, 1931.

- [87] A.J. Lotka, *The extinction of families, II*, J. Washington Acad. Sci. **21**, pp 453-459, 1931.
- [88] J.W. Lyklema and C. Evertsz, *The Laplacian Random Walk*, in [99], pp 87-92.
- [89] J. Maynard Smith, *The Evolution of Sex*, Cambridge: Cambridge University Press, 1978.
- [90] J. Maynard Smith, *Hypercycles and the origin of life*, Nature **280**, pp 445-446, 1979.
- [91] J. Maynard Smith, *Evolution and the Theory of Games*, Cambridge: Cambridge University Press, 1982.
- [92] J. Maynard Smith, *The causes of extinction*, in *Evolution and Extinction*, W.G. Chaloner *et al.*, editors, The Royal Society, London, 1989.
- [93] J. Maynard Smith and J. Haigh, *The hitch-hiking effect of a favourable gene*, Genet. Res., Camb. **23**, pp 23-35, 1974.
- [94] M. Mézard, G. Parisi and M. A. Virasoro, *Spin-Glass Theory and Beyond*, Singapore: World Scientific, 1987.
- [95] Ernest Nagel, *The Structure of Science*, New York: Harcourt, Brace & World, 1961.
- [96] N.H. Packard, *Intrinsic Adaption in a Simple Model for Evolution*, in [82], pp 141-155.
- [97] R. Pearson, *Climate and Evolution*, Academic Press, London 1978.
- [98] L. Peliti, *Random Walks with Memory*, in [99], pp 73-81.
- [99] L. Pietronero and E. Tosatti (Eds.), *Fractals in Physics*, North Holland, Amsterdam, 1986.
- [100] E. Rasmusen, *Games and Information*, Blackwell, Oxford, 1989.
- [101] S. Rasmussen, C. Knudsen, R. Feldberg, and M. Hindsholm, *The Coreworld: Emergence and Evolution of Cooperative Structures in a Computational Chemistry*, Physica **D42**, pp 111-134, 1990.
- [102] S. Rasmussen, C. Knudsen, and R. Feldberg, *Dynamics of Programmable Matter*, in [84], pp 211-254.
- [103] D.M. Raup, *Biological Extinction in Earth History*, Science **231**, pp 1528-1533, 1986.
- [104] I. Rechenberg, *Evolutionstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Frommann-Holzboog, Stuttgart-Bad Cannstatt, 1973.
- [105] C.W. Reynolds, *Flocks, Herds, and Schools: A Distributed Behavioral Model (Proceedings of SIGGRAPH '87)*, Computer Graphics **21**, pp 25-34, 1987.
- [106] Th.W. Ruijgrok and E.G.D. Cohen, *Deterministic Lattice Gas Models*, Phys. Lett. **A 133**, pp 415-418, 1988.
- [107] H.G. Schuster *Deterministic Chaos*, VCH Verlagsgesellschaft, Weinheim, 1988.
- [108] M. Serva and L. Peliti, *A statistical model of an evolving population with sexual reproduction*, J. Phys. **A24**, pp L705-L709, 1991.
- [109] C.E. Shannon and W. Weaver, *The Mathematical Theory of Communication*, University of Illinois Press, Urbana, Illinois, 1949.
- [110] F.L. Spitzer, *Principles of Random Walks*, Springer-Verlag, Berlin Heidelberg, 1976.
- [111] J.F. Steffensen, *Om sandsynligheden for at afkommet uddør*, Matematisk Tidsskrift B **1**, pp 19-23, 1930.

-
- [112] J.F. Steffensen, *Deux problèmes du calcul des probabilités*, Ann. Inst. H. Poincaré **3**, pp 319-344, 1932.
- [113] C. Strobeck, J. Maynard Smith, and B. Charlesworth, *The effects of the hitch-hiking on a gene for recombination*, Genetics **82**, pp 547-558, 1976.
- [114] L. Van Valen, *A new evolutionary law*, Evolutionary Theory **1**, pp 1-30, 1973.
- [115] T. Vicsek, *Fractal Growth Phenomena*, World Scientific, Singapore, 1989.
- [116] H.W. Watson, *Solution to Problem 4001*, Educational Times, 1 August, 1873.
- [117] G.M. Werner and M.G. Dyer, *Evolution of Communication in Artificial Organisms*, in [84], pp 659-687.
- [118] S. Wolfram (Ed.), *Theory and Applications of Cellular Automata*, World Scientific, Singapore, 1986.
- [119] R. Wolter, *Selbstorganisierende Evolutionsmodelle*, Diplomarbeit WU D 92-12, Wuppertal, 1992.
- [120] R. Wolter, Private Mitteilungen.
- [121] J. Zhuo, G. Murthy and S. Redner, preprint 1991.

Danksagung

An erster Stelle möchte ich Herrn Prof. Dr. Peter Grassberger danken. Er initiierte das Bit-String-Modell und förderte die Arbeit durch viele anregende und hilfreiche Diskussionen. Obwohl die Forschungen nicht ganz die von ihm erhoffte Richtung nahmen, unterstützte er mich doch stets bei meinen Versuchen, das Thema so weit wie möglich auszuloten.

Seinen Bemühungen habe ich es auch zu verdanken, daß ich in den Jahren 1990 und 1991 am Workshop *Complexity and Evolution* des *Institute for Scientific Interchange* (I.S.I.) in Turin teilnehmen konnte. Diese Aufenthalte wurden von der ISI finanziert. Sie haben mir Kontakte zu anderen Wissenschaftlern ermöglicht, die ebenfalls auf dem Gebiet Artificial Life arbeiten. Hier seien vor allem Dr. Walter Fontana und Prof. Dr. Norman H. Packard genannt. Beide erlaubten mir, ihre Quelltexte einzusehen und ihre Programme zu testen. Aber auch die anderen Teilnehmer (von denen hier nur einige wenige namentlich erwähnt werden: Richard Bagley, Aviv Bergman, Doyne Farmer; Andrew Assad, Joe Breeden, E. Atlee Jackson) waren immer zu Diskussionen bereit. Dies war sehr wichtig für mich, da ich dort mein Modell vorstellen konnte und falls nötig verteidigen mußte.

Prof. Grassberger und Dr. Fontana unterstützten mich auch bei der Bewerbung um ein Stipendium der Volkswagen-Stiftung im Schwerpunktbereich Synergetik. Dieses Stipendium (AZ I/66 995), das von März 1991 bis August 1992 gelaufen ist, hat es mir in den letzten anderthalb Jahren erlaubt, mich ganz den Forschungen an dieser Arbeit zu widmen.

Mein weiterer Dank gilt Robert Wolter, bei Diskussionen mit ihm sind mir viele Dinge klarer geworden, weil sie mir geholfen haben, meine Gedanken besser zu formulieren. Außerdem half er mir, indem er einige Programmierwerkzeuge schuf. Seine Abwandlung des Bit-String-Modells, das Energieflußmodell, beweist außerdem, daß die in dieser Arbeit gefundenen Resultate allgemeinere Gültigkeit besitzen. Außerdem hat er diese Arbeit kritisch auf Fehler durchgesehen.

Die Abbildungen in dieser Arbeit wurden mit verschiedenen Programmen erstellt, die POSTSCRIPT-Dateien erstellen. Diese können dann direkt in die L^AT_EX-Datei eingebunden werden. Die meisten Plots wurden mit GNU PLOT (einem Public Domain Programm) und **c3** (Autor Andreas Haase) erzeugt.

Mein Dank gilt auch meinen Kollegen und Freunden: Astrid Althaus, Almut König, Wiebke Krüger, Julia Niewöhner, Susanne Orbach, Sonja Schüssler, Martina Steininger, Tony Burkitt, Ralph Ebner, Gerald Fahner, Rainer Hegger, Ingo Hoffmann, Holger Kantz, Michael Karbach, Jörg-Uwe Klaetke (†), Thomas Lippert, Andreas Pohlmann, Arkadi Pikovsky, Carsten Schaffrath, Michael Schmidt, Manfred Schürmann, Stefan Schwenke, Norbert Tellmann, Peer Ueberholz. Besonders meine Bürokollegen haben eine sehr angenehme Arbeitsatmosphäre ermöglicht.

Und last but not least möchte ich mich auch bei meiner Familie für ihre immerwährende Unterstützung bei meinem Studium bedanken.