

**Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig**

**Schemen und Rekombinationsverfahren in
Genetischen Algorithmen insbesondere bei
Vorliegen von Symmetrien**

by

Susanne Schindler

Preprint no.: 99

2004



Universität Leipzig
Fakultät für Mathematik und Informatik
Mathematisches Institut
Augustusplatz 10/11
04109 Leipzig

Max-Planck-Institut für Mathematik in den Naturwissenschaften
Inselstraße 22
04103 Leipzig

Diplomarbeit

Schemen und Rekombinationsverfahren in Genetischen Algorithmen insbesondere bei Vorliegen von Symmetrien

Susanne Schindler
geb. 22.2.1979
Studiengang Mathematik

Leipzig, den 14. Oktober 2004

Betreuer:
Prof. Dr. rer. nat. Jürgen Jost

Danksagung

Ich bedanke mich herzlich bei Prof. Jost (MPI MIS Leipzig) für die sehr gute Zusammenarbeit. Ein weiteres Dankeschön geht an die stets konstruktiv kritischen Korrekturleser Konrad Kaltenbach (Uni Leipzig) und M. Sc. Holger Arnold (Uni Potsdam).

Susanne Schindler

Kurzzusammenfassung

Diese Arbeit geht aus der Diplomarbeit hervor, die ich zum Abschluss meines Studiums der Wirtschaftsmathematik an der HTWK Leipzig eingereicht habe. Die damalige Arbeit entstand unter der Betreuung von Prof. Jost (MPI MIS Leipzig) und Prof. Rudolph (HTWK Leipzig). Der Titel lautete „Schemen in Genetischen Algorithmen“.

Die ersten drei Kapitel sind, abgesehen von einigen Änderungen und Ergänzungen, aus der ersten Diplomarbeit übernommen. Die Kapitel 4 bis 8 sind vollständig neu erarbeitet.

Kernpunkte der ersten Diplomarbeit waren die mathematische Formulierung des Genetischen Algorithmus und das Aufstellen der Evolutionsgleichung für Schemen. Diese Gleichung beschreibt die quantitative Entwicklung der Schemen in der Population. In der vorliegenden Arbeit wird gezeigt, dass diese Evolutionsgleichung zu der Gleichung von C. R. Stephens äquivalent ist.

Weiterhin werden die Untersuchungen auf Genetische Algorithmen ausgeweitet, die eine Symmetrie-Fitnessfunktion und einen gerichteten Rekombinationsoperator verwenden. In diesen Algorithmen werden die Individuen bezüglich ihrer Symmetrie im Genom bewertet und die Kreuzung nur für genotypisch ähnliche Individuen zugelassen. Hierfür wird die angepasste Evolutionsgleichung hergeleitet.

Als weiterer Bestandteil wird das Konzept der „Effektiven Fitness“ vorgestellt. Dieses liefert ein realistisches Maß für die Fortpflanzungschancen der Populationselemente.

Verschiedene Basen für den Suchraum und seinen Dualraum, wie zum Beispiel die Stringbasis und Bausteinbasis, werden erläutert. Abgesehen von der Operatorenschreibweise können die genetischen Operatoren auch als Tensoren dargestellt werden. Die Renormierungsgruppe wird als Lösungsmethode der Evolutionsgleichung vorgestellt.

Den Abschluss bilden numerische Untersuchungen am Genetischen Algorithmus mit Symmetrie-Fitnessfunktion und gerichtetem Rekombinationsoperator.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Einführung in Evolutionäre Algorithmen	1
2	Grundlagen	7
2.1	Der einfache genetische Algorithmus	7
2.2	Einführung in die Schematheorie	15
2.3	Zusammenfassung	17
3	Hollands Schemensatz und die Evolutionsgleichungen	19
3.1	Hollands Schemensatz	19
3.2	Modifikation von Hollands Schemensatz	24
3.3	Stephens Evolutionsgleichung	33
3.4	Verallgemeinerung der Evolutionsgleichung auf beliebige genetische Operatoren	34
3.5	Homogenisierung der Population	36
3.6	Zusammenfassung	37
4	Die Symmetrie als Fitnessfunktion und die gerichtete Rekombination	39
4.1	Die Fitnesslandschaft	40
4.2	Wirkung der Mutation in der Fitnesslandschaft	41
4.3	Wirkung der Rekombination in der Fitnesslandschaft	45
4.4	Evolutionsgleichung für die neue Fitnessfunktion	52
4.5	Der gerichtete Rekombinationsoperator	53
4.6	Evolutionsgleichung für die neue Fitnessfunktion und die gerichtete Rekombination	57
4.7	Zusammenfassung	58
5	Effektive Fitness	59
6	Basen des Suchraums	67
6.1	Der Suchraum und sein Dualraum	67
6.2	Basen von $Fun(\mathfrak{A})$	69
6.3	Basiswechsel	76
6.4	Die Basen des Dualraums $Fun^*(\mathfrak{A})$	87
6.5	Transformation der Wahrscheinlichkeitsvektoren	87
6.6	Tensorschreibweise der genetischen Operatoren	88
6.7	Die Wirkung der genetischen Operatoren bezüglich der verschiedenen Basen	92

Inhaltsverzeichnis

6.8	Zusammenhang zwischen Bausteinbasis und Renormierungsgruppe	95
6.9	Zusammenfassung	98
7	Simulationsergebnisse	99
7.1	Die Mutation	99
7.2	Die Rekombination	101
7.3	Die Fitnessfunktion	107
8	Zusammenfassung	111
	Bezeichnungen	113
	Literaturverzeichnis	117

1 Einleitung

Im Zeitalter der Bioinformatik erscheint die Synthese von Fachbegriffen aus grundverschiedenen Wissenschaften nicht mehr befremdlich. So kann man erahnen, was sich hinter „Genetischen Algorithmen“ verbirgt. Inhalt dieses Kapitels soll die Präzisierung dieser Ahnung sein. Grundbegriffe und das Konzept der Genetischen Algorithmen werden vorgestellt. Die Einführung ist bewusst auf das Wesentliche reduziert. Ausführlichere Beschreibungen zum Thema „Genetische Algorithmen“ können in [Gol89], [Mic96], [Mit96], [Nis97] oder [Poh00] nachgelesen werden.

1.1 Einführung in Evolutionäre Algorithmen

Genetische Algorithmen werden zur Lösung von Optimierungsproblemen verwendet. Ein Optimierungsproblem besteht in der Minimierung oder Maximierung einer Funktion von mehreren Variablen. Diese Funktion wird *Zielfunktion* genannt. Die Menge aller möglichen Variablenbelegungen wird als *Suchraum* bezeichnet. Bei einem Optimierungsproblem können eine oder mehrere Bedingungen an die Variablen oder die Zielfunktionswerte gestellt werden. Als traditionelle Optimierungsverfahren werden in dieser Arbeit die deterministischen Verfahren verstanden. Sie sind zumeist stark auf einen bestimmten Fall der Optimierung spezialisiert. Dazu gehören zum Beispiel der Simplexalgorithmus, das Newton-Verfahren, Varianten der Quasi-Newton-Verfahren und Trust-Region-Verfahren. Bei diesen Verfahren werden Stetigkeit, Differenzierbarkeit oder Konvexität der Zielfunktion vorausgesetzt. Sie bestimmen in der Regel entweder eine Folge von Variablenbelegungen, die gegen das Optimum konvergieren soll oder bestimmen durch Transformation des ursprünglichen Optimierungsproblems zu einem anderen das Optimum des Ausgangsproblems.

Evolutionäre Algorithmen Genetische Algorithmen bilden eine Untergruppe der Evolutionären Algorithmen. Letztere werden seit den 60er Jahren als Alternative zu traditionellen Optimierungsverfahren diskutiert, weiterentwickelt und mathematisch untersucht.

In der Literatur findet sich derzeit keine einheitliche Definition von Evolutionären Algorithmen. Die meisten Veröffentlichungen stimmen darin überein, dass Evolutionäre Algorithmen stochastische Verfahren sind. Sie bieten Lösungsansätze für Optimierungsprobleme, indem sie bestimmte Elemente aus der Evolutionstheorie modellieren.

Evolutionäre Algorithmen basieren ähnlich der darwinschen Lehre auf dem Prinzip „Der Stärkere überlebt“. Das Ziel ist, Lösungen in einer Pro-

1 Einleitung

grammumgebung selbstständig entwickeln zu lassen, indem man die der Evolution zugrundeliegend geglaubten Prozesse (auch *Evolutionsfaktoren* genannt) modelliert.

Die zentralen Evolutionsfaktoren sind die Vererbung und Vermischung des Erbguts, die Selektion der am besten an die Umwelt angepassten Individuen und die Mutation.

Evolutionäre Algorithmen arbeiten mit einer *Population* P_t , deren Elemente *Individuen* genannt werden. Diese Individuen repräsentieren Elemente des Suchraums, also konkrete Variablenbelegungen. Dabei können zwei oder mehr Individuen auch dieselben Belegungen repräsentieren. Die Individuen werden mit den Elementen, die sie repräsentieren, identifiziert.

Durch Anwendung der *genetischen Operatoren* auf die Population P_t unterliegt diese Population generationsähnlichen Veränderungen. Dies bedeutet, dass die Individuen von P_{t+1} aus Transformationen der Individuen von P_t hervorgehen. Der Index $t \in \mathbb{N}_0$ der Population P_t bezeichnet dabei die Generation, in der sich die Population befindet. Genetische Operatoren modellieren die oben genannten Evolutionsfaktoren. Zu den elementarsten genetischen Operatoren gehören die Fitnesszuweisung, die Selektion, die Rekombination und die Mutation. Sie werden später in diesem Abschnitt näher beschrieben.

Ablauf eines evolutionären Algorithmus Der Ablauf besteht aus dem aufeinander folgenden Anwenden der genetischen Operatoren. Die Reihenfolge der genetischen Operatoren kann variieren. Hier wird die Reihenfolge vorgestellt, die in dieser Arbeit verwendet wird. Abbildung 1.1 zeigt den typischen Ablauf eines Evolutionären Algorithmus.

Ausgehend von einer Anfangspopulation P_0 werden Folgepopulationen P_{t+1} in folgender Weise gebildet: Nach einer Fitnesszuweisung werden einige Individuen ausgewählt (Selektionsoperator), um Klone, also exakte Kopien ihrer selbst, zu erhalten. Die Auswahl dieser Individuen kann auf Grund der Erfüllung der an die Variablen gestellten Bedingungen erfolgen, oder durch den Zielfunktionswert begründet sein. Anstelle von „Klonen“ spricht man häufiger von „*Nachkommen*“. Diese Nachkommen werden durch den Rekombinationsoperator und den Mutationsoperator transformiert. Im Anschluss werden die Individuen der Folgepopulation P_{t+1} aus den Individuen von P_t und den transformierten Nachkommen bestimmt. Auf welche Art die Folgepopulation zusammengesetzt wird und auf welche Weise die Transformationen des Rekombinations- und Mutationsoperators durchgeführt werden, ist problemabhängig. Die ursprüngliche Population wird dann durch P_{t+1} ersetzt. Der Zyklus vom Anwenden des Selektionsoperators bis zum Ersetzen der Population wird als *Generation* bezeichnet. Anschließend wird wieder die Fitness zugewiesen, selektiert, rekombiniert und mutiert. Dies geschieht so lange, bis ein bestimmtes Abbruchkriterium erfüllt ist. Das kann eine maximale Generationenanzahl sein oder eine ausbleibende signifikante Änderung des besten Zielfunktionswertes eines Individuums in der Population über mehrere Generationen. Durch das

1.1 Einführung in Evolutionäre Algorithmen

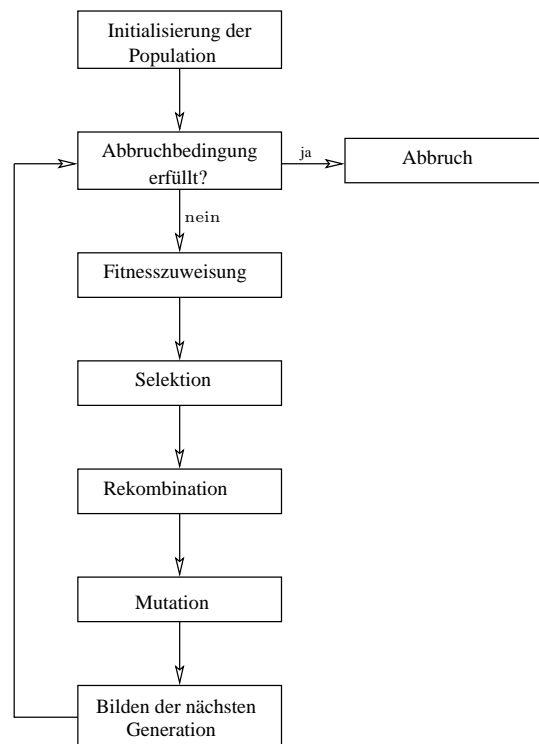


Abbildung 1.1: Beispiel für den Ablauf eines Evolutionären Algorithmus

wiederholte Anwenden der genetischen Operatoren erhofft man sich eine schrittweise Annäherung der Individuen an das Optimum oder die Optima des zugrundeliegenden Problems.

Abbildung 1.2 gibt einen Überblick zur Einordnung und liefert eine Zusammenfassung der gebräuchlichsten Arten der Evolutionären Algorithmen. Unabhängig voneinander wurden zwei Gruppen von Algorithmen entwickelt, welche Evolutionsprozesse simulieren. Beide Gruppen werden heute als Evolutionäre Algorithmen zusammengefasst.

Genetische Algorithmen Für Genetische Algorithmen existiert ebensowenig wie für Evolutionäre Algorithmen eine einheitliche Definition. Genetische Algorithmen werden als Verfahren betrachtet, welche mit binären Zeichenketten arbeiten und eine Form der Selektion, der Rekombination und der Mutation verwenden. Die Verwendung der Rekombination ist der wesentliche Unterschied zu Evolutionären Algorithmen. Da die Rekombination sehr wirksam bei der Optimierung ist, findet man sie mittlerweile auch bei anderen Evolutionären Algorithmen. Ursprünglich zeichnete jedoch die Rekombination den Genetischen Algorithmus aus.

Die Bezeichnungen der Objekte eines genetischen Algorithmus sind an die Biologie angelehnt. So arbeitet ein Genetischer Algorithmus mit *Genomen*, *Genen* und *Allelen*. In natürlichen Zellkernen ist die Erbinformation im Genom gespeichert, ein Gen entspricht einem Genomabschnitt und die

1 Einleitung

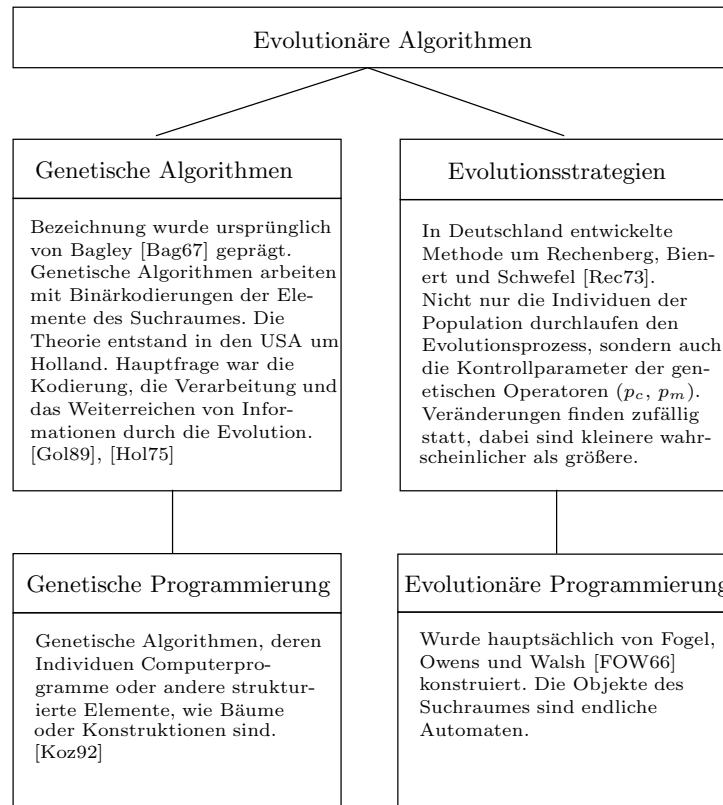


Abbildung 1.2: Evolutionäre Algorithmen

verschiedenen möglichen Ausprägungen eines Gens werden Allele genannt. Bei Genetischen Algorithmen werden die binären Zeichenketten als Genome oder als *Genotypen* bezeichnet. Die Gene kodieren einzelne Eigenschaften der Elemente des Suchraumes und können mehrere Zeichenpositionen vereinigen. In dieser Arbeit werden der Einfachheit halber die einzelnen Zeichenpositionen als Gene und ihre Bitbelegung als Allel bezeichnet.

Im Folgenden sollen die genetischen Operatoren genauer betrachtet werden. Es wird sich auf Operatoren beschränkt, die auf binären Zeichenketten agieren können, und somit für den Gebrauch in Genetischen Algorithmen geeignet sind.

Fitnesszuweisung Die Fitnesszuweisung erfolgt in der Regel vor der Selektion. Jedem Individuum wird ein Fitnesswert zugewiesen. Dieser ist abhängig von seinem Zielfunktionswert und der Erfüllung oder Nichterfüllung der Bedingungen, die an die Variablen gestellt werden. Der Fitnesswert ist ein Maß für die Anzahl an Nachkommen, die das Individuum bekäme, wenn es selektiert würde.

In dieser Arbeit wird hauptsächlich die *proportionale Fitnesszuweisung* benutzt. Sie findet vor der Selektion statt. Die Fitness eines Individuums wird proportional zu seinem Zielfunktionswert bestimmt.

Die Fitnesszuweisung kann auch nach der Mutation erfolgen. Dann ist

die Fitness eines Individuums ein Maß dafür, wie wahrscheinlich es ist, dass es in der Population ersetzt oder in die Folgepopulation eingefügt wird.

Selektion Die Selektion kann praktisch an jeder beliebigen Stelle im Algorithmus erfolgen. Sie legt fest, welche Individuen tatsächlich Nachkommen erhalten oder welche Individuen durch andere in der Folgepopulation ersetzt oder eingefügt werden.

Die in dieser Arbeit gebrauchte Selektionsform ist die *fitnessproportionale Selektion*. Sie bestimmt, welche Individuen zur Rekombination ausgewählt werden. Die Wahrscheinlichkeit, dass ein Individuum ausgewählt wird, entspricht seiner relativen Fitness. Diese ergibt sich aus dem Verhältnis seines Fitnesswertes zu der Summe der Fitnesswerte aller Populations-elemente.

Rekombination Die Rekombination wirkt auf den Genomen von mindestens zwei Individuen und wird meist in einer Form des Überkreuzungsaustausches (auch „Crossover“ oder „Kreuztausch“ genannt) genutzt. Das Ziel der Rekombination ist das Einbringen von neuen Individuen in die Population sowie die Kombination zweier guter Individuen zu möglichst noch besseren Nachkommen. Häufig wird der *n-Punkt Kreuztausch* mit $n = 1$ verwendet. Mit der Kreuztauschwahrscheinlichkeit p_c werden gemäß einer gleichmäßigen Verteilung n Schnittstellen zwischen zwei Genen zweier Genome ausgewählt und die dazwischenliegenden Abschnitte ausgetauscht.

Mutation Die Mutation operiert auf den Zeichenketten einzelner Individuen in der Regel nach dem Rekombinationsschritt. Der Zweck der Mutation liegt in der Vergrößerung der Diversität der Population, denn als einziger genetischer Operator ändert die Mutation die Allelenhäufigkeit. Ist ein Allel an einem bestimmten Gen in allen Genomen verloren, so kann dieses Allel nur durch die Mutation wieder in die Population gelangen. Bei der *bitweisen Mutation* wird jedes Allel mit der Mutationswahrscheinlichkeit p_m negiert, das heißt eine „0“ durch eine „1“ ersetzt und umgekehrt.

Vorteile und Nachteile von Evolutionären Algorithmen Einige Unterschiede zwischen Evolutionären Algorithmen und traditionellen Optimierungsverfahren, und zwar der stochastische Charakter und die Verwendung von Populationen, wurden bereits angesprochen. Ein weiterer wesentlicher Unterschied besteht darin, dass Evolutionäre Algorithmen häufig mit Kodierungen der Elemente aus dem Suchraum arbeiten.

Durch das einfache und wenig spezialisierte Konzept besitzen Evolutionäre Algorithmen breite Anwendungsmöglichkeiten. Sie finden nicht immer die Optimallösung. Dieser Punkt hatte lange Zeit die Forscher beschäftigt, bis man herausfand, dass die Zielstellungen von Optimierern und Evolutionären Algorithmen nicht immer identisch sind. So ist man als Anwender am Optimum des Problems interessiert, ein Evolutionärer

1 Einleitung

Algorithmus ist jedoch so konstruiert, dass er das Optimum der Evolution sucht. Das heißt es wird ein Element gesucht, das nicht nur die Optimierungsaufgabe gut löst, sondern auch noch möglichst robust gegenüber der Rekombination und Mutation ist. Darauf wird in Kapitel 5 eingegangen.

Ein Vorteil von Evolutionären Algorithmen liegt in dem relativ schnellen Finden einer Lösung. Besonders geeignet sind sie daher bei Optimierungsproblemen, bei denen möglichst rasch eine Lösung gefunden werden soll, es aber ausreicht, wenn ein Nebenmaximum gefunden wird, das fast genauso gut wie das exakte Optimum ist.

Sie eignen sich weniger, wenn die Auswertung der Zielfunktion besonders zeitintensiv ist, denn die Verwendung von Populationen erfordert zahlreiche Zielfunktionswertberechnungen.

Die populationsbasierte Vorgehensweise bietet den Vorteil, dass mehrere Optima bei multimodalen Zielfunktionen gefunden werden können. Da keinerlei Informationen über die Zielfunktion, abgesehen von den Funktionswerten, benötigt werden und auch keine Bedingungen an die Zielfunktion gestellt werden, eignen sich Evolutionäre Algorithmen auch bei nichtlinearen und unstetigen Zielfunktionen.

2 Grundlagen

Mit diesem Kapitel werden Bezeichnungen und Konstrukte eingeführt, die für die Formalisierung und die Analyse von Genetischen Algorithmen notwendig sind. In Abschnitt 2.1 werden die genetischen Operatoren konkretisiert. Durch diese Konkretisierung der genetischen Operatoren erhält man einen speziellen Genetischen Algorithmus. Dieser wird ausgezeichnet und „einfacher genetischer Algorithmus“ genannt.

In der Forschung werden verschiedene Modellierungsansätze für Genetische Algorithmen verfolgt. Zum Beispiel modelliert Vose [Vos99] die Population, auf der die genetischen Operatoren wirken, als dynamisches System. Andere Ansätze nutzen Markov-Prozesse oder Methoden der Statistischen Mechanik. Diese Arbeit beschäftigt sich mit der Schematheorie, welche J. H. Holland entwickelt hat, um ein Verständnis für die Wirkungsweise von Genetischen Algorithmen zu erhalten [Hol75]. Mit dem Schemensatz zeigt Holland, dass sich bestimmte Klassen von Individuen mit wünschenswerten Eigenschaften exponentiell schnell in der Population ausbreiten und versucht damit, eine Erklärung für das schnelle Finden einer hinreichend guten Lösung zu geben. In Abschnitt 2.2 wird die Schematheorie vorgestellt, damit später der Schemensatz von Holland hergeleitet werden kann.

2.1 Der einfache genetische Algorithmus

Genetische Operatoren wurden bereits in Abschnitt 1.1 vorgestellt. Sie modellieren die aus der Evolutionstheorie übernommenen Evolutionsfaktoren. Genetische Operatoren können beliebig variiert und an das zu untersuchende Problem angepasst werden. Daher ist es notwendig, sich vor Beginn der Untersuchungen auf ein konkretes Modell eines genetischen Algorithmus zu einigen. In dieser Arbeit wird die proportionale Fitnesszuweisung verwendet sowie die Selektion der am besten an die Umwelt angepassten Individuen (fitnessproportionale Selektion), die einfachste Form des Kreuztauses (1-Punkt Kreuztausch) und die Mutation auf binären Zeichenketten (bitweise Mutation). Der resultierende Algorithmus wird einfacher genetischer Algorithmus genannt.

In dieser Arbeit wird er nur für Optimierungsprobleme genutzt, welchen eine Maximierungsaufgabe einer reellwertigen nichtnegativen Zielfunktion z zu Grunde liegt.

Die Länge der Genome wird mit l bezeichnet. Die Populationsgröße soll gerade sein, um den 1-Punkt Kreuztausch zwischen jeweils zwei Individuen durchführen zu können. Die Populationsgröße wird mit $2n$ ($n \in \mathbb{N}$)

2 Grundlagen

bezeichnet. Weiterhin werden die Individuen nicht nur mit den Elementen des Suchraums, welche sie repräsentieren, sondern auch mit ihren Genomen identifiziert. Die Genomschreibweise für ein Individuum x ist $x = (x^1, x^2, \dots, x^l) = x^1 x^2 \dots x^l$.

Um den einfachen genetischen Algorithmus von dem mathematischen Verfahren, welches dem Algorithmus zu Grunde liegt und hier formuliert werden soll, zu unterscheiden, wird dem Verfahren der Name „genetisches Verfahren“ gegeben. Ein einfacher genetischer Algorithmus soll eine Implementierung des genetischen Verfahrens sein. An dieser Stelle wird noch einmal daran erinnert, dass es sich um ein stochastisches Verfahren handelt. In der recherchierten Literatur befinden sich keine, für die Zwecke dieser Arbeit, passenden mathematischen Beschreibungen eines genetischen Algorithmus. Daher werden im Folgenden Abbildungen eingeführt, welche als Grundlage für die genetischen Operatoren des einfachen genetischen Algorithmus dienen können.

Das genetische Verfahren konstruiert, ausgehend von einer Population P_0 eine Folge von Populationen $\{P_t\}_{t \geq 1}$, also Mengen von Elementen aus dem Suchraum \mathfrak{A} , in denen ein Element mehrfach auftreten kann. Die Folgepopulation wird erzeugt, indem die einzelnen Abbildungen nacheinander auf die Population P_t angewendet werden.

Eine Population ist eine ungeordnete $2n$ -elementige Menge mit Elementen aus \mathfrak{A} . Zunächst wird \mathbb{P} als die Menge aller möglichen Populationen der Größe $2n$ definiert:

$$\mathbb{P} = \hat{\mathfrak{A}}^{2n}.$$

Dabei ist $\hat{\mathfrak{A}}^{2n}$ die Faktorgruppe oder die Menge der Äquivalenzklassen zu der Äquivalenzrelation \sim :

$$\hat{\mathfrak{A}}^{2n} = \mathfrak{A}^{2n} / \sim.$$

Für die Äquivalenzrelation gilt:

$$(a_1, \dots, a_{2n}) \sim (a'_1, \dots, a'_{2n}) \Leftrightarrow \begin{aligned} \exists \pi \in S_{2n} : a'_k &= a_{\pi(k)} \\ \forall k \in \{1, \dots, 2n\} \quad \text{und} \quad a_k, a'_k &\in \mathfrak{A}, \end{aligned}$$

wobei S_{2n} die symmetrische Gruppe der Ordnung $2n$ ist. Nun wird eine Funktion $P : \mathbb{N}_0 \rightarrow \mathbb{P}$ definiert, die jedem Zeitschritt t eine Population aus \mathbb{P} zuordnet:

$$P(t) =: P_t.$$

Nun sind \mathbb{P} und P_t genau definiert und können zur Beschreibung der genetischen Operatoren genutzt werden.

Fitnesszuweisung Der Operator der proportionalen Fitnesszuweisung wird folgendermaßen definiert:

Definition 2.1 Die proportionale Fitnesszuweisung ist eine Abbildung F vom Suchraum \mathfrak{A} in die Menge der positiven reellen Zahlen, die folgendermaßen beschrieben ist:

$$F : \mathfrak{A} \longrightarrow \mathbb{R}_0^+ \\ F(x) = \begin{cases} a z(x) & : x \in D(z) \\ 0 & : \text{sonst} \end{cases} \quad a > 0.$$

Der Wert $F(x)$ heißt *Fitnesswert* des Elementes $x \in \mathfrak{A}$. Die proportionale Fitnesszuweisung ordnet jedem Element aus \mathfrak{A} einen Fitnesswert zu, welcher gleich dem Zielfunktionswert von x multipliziert mit einem Proportionalitätsfaktor a ist. Durch die Beschränkung auf Maximierungsprobleme und die Definition von F ist gesichert, dass jedes Element einen nichtnegativen Fitnesswert besitzt. Der Fitnesswert ist unabhängig von der Generation t .

Der durchschnittliche Fitnesswert der Population P_t wird mit \bar{F}_t bezeichnet:

$$\bar{F}_t = \frac{1}{2n} \sum_{x \in P_t} F(x).$$

Definition 2.2 Eine Funktion

$$f_t : \mathfrak{P}(P_t) \longrightarrow \mathbb{R}[0, 1]$$

heißt *Fitnessfunktion* zu einer Abbildung $F : \mathfrak{A} \longrightarrow \mathbb{R}_0^+$, wenn gilt:

$$(a) \quad f_t(\{x\}) =: f_t(x) = \begin{cases} \frac{1}{2n} & : \text{wenn } F(y) = 0 \forall y \in P_t \\ \frac{F(x)}{\sum_{y \in P_t} F(y)} & : \text{wenn } \exists y \in P_t : F(y) > 0 \end{cases}$$

$$(b) \quad f_t\left(\bigcup_{i \in I} A_i\right) = \sum_{i \in I} f_t(A_i) \quad \text{für paarweise disjunkte } A_i \in \mathfrak{P}(P_t).$$

Der Wert $f_t(x)$ wird *relative Fitness* oder auch *relativer Fitnesswert* von x genannt und ist von P_t abhängig.

Selektion Für die Beschreibung der fitnessproportionalen Selektion sind mehrere Schritte notwendig. Zum einen wird eine Bijektion $n_t^1 : P_t \longrightarrow \{1, 2, \dots, 2n\} =: \Omega$ definiert, die dazu dient, die Individuen von P_t durchnummerieren, und zum anderen wird ein Wahrscheinlichkeitsraum konstruiert.

2 Grundlagen

Hilfssatz 2.1 Das Tupel $(\Omega, \mathfrak{P}(\Omega), f_t)$ beschreibt einen Wahrscheinlichkeitsraum.

Beweis: Es muss Folgendes gezeigt werden:

a) $\mathfrak{P}(\Omega)$ ist eine σ -Algebra über Ω .

b) f_t ist ein Wahrscheinlichkeitsmaß.

zu a) $\mathfrak{P}(\Omega)$ ist bekanntlich eine σ -Algebra über Ω .

zu b) aus [WL92]:

„Ist Ω eine Ergebnismenge und ist \mathfrak{A} eine σ -Algebra von Ereignissen über Ω , so heißt eine Abbildung $P : \mathfrak{A} \rightarrow \mathbb{R}$ ein *Wahrscheinlichkeitsmaß*, wenn gilt:

(i) $P(A) \geq 0$ für alle $A \in \mathfrak{A}$,

(ii) $P(\Omega) = 1$,

(iii) $P\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} P(A_i)$ für paarweise unvereinbare Ereignisse $A_i \in \mathfrak{A}$.“

Nach Definition von f_t sind (i) und (iii) erfüllt.

Zu (ii): Die Abbildung n_t^1 ist eine Bijektion $n_t^1 : P_t \rightarrow \Omega$. Somit kann geschrieben werden:

$$f_t(\Omega) = f_t(P_t) = f_t\left(\bigcup_{x \in P_t} x\right) \stackrel{(iii)}{=} \sum_{x \in P_t} f_t(x).$$

Nun sind zwei Fälle auf Grund der Definition von f_t zu unterscheiden:

1. Fall: $F(x) = 0 \quad \forall x \in P_t$, dann ist

$$\sum_{x \in P_t} f_t(x) = \sum_{x \in P_t} \frac{1}{2n} = 2n \frac{1}{2n} = 1, \quad \text{da } |P_t| = 2n.$$

2. Fall: $\exists x \in P_t : F(x) > 0$, dann ist

$$\sum_{x \in P_t} f_t(x) = \sum_{x \in P_t} \frac{F(x)}{\sum_{y \in P_t} F(y)} = \frac{\sum_{x \in P_t} F(x)}{\sum_{y \in P_t} F(y)} = 1.$$

Somit ist (ii) erfüllt und der Hilfssatz bewiesen. \square

Mit Hilfe der Bijektion n_t^1 kann eine Zufallsgröße $X_t : \Omega \rightarrow \Omega \subset \mathbb{R}$ über dem Wahrscheinlichkeitsraum $(\Omega, \mathfrak{P}(\Omega), f_t)$ definiert werden:

$$\begin{aligned} X_t(\omega) &= n_t^1(x), \text{ wenn das Individuum } x \text{ ausgewählt wurde} \\ &= n_t^1(x) \text{ mit der Wahrscheinlichkeit } f_t(x), x \in P_t, \omega \in \Omega. \end{aligned}$$

2.1 Der einfache genetische Algorithmus

Das Ergebnis eines zufälligen Experimentes ist ω . Wird das Zufallsexperiment für X_t genau $2n$ -mal wiederholt, so bezeichnet $X_t(\omega_i)$, $\omega_i \in \Omega$, das Ereignis, welches im i -ten Versuch eingetreten ist.

Der Zweck der Selektion für das genetische Verfahren liegt in der Auswahl der Individuen für den Rekombinationsoperator, der auf jeweils zwei Individuen wirkt. Insofern wird der Prozess des Auswählens und der anschließenden Paarbildung abgekürzt und somit auch die Rechnungen vereinfacht. Realisiert wird dies durch die Anordnung zu Paaren während der Selektion. Das Versuchsergebnis ist somit eine Menge von Paaren, in denen jedes Element x mit der Wahrscheinlichkeit $f_t(x)$ auftritt. Diese Menge wird mit S_t bezeichnet:

$$S_t = \left\{ \{X_t(\omega_1), X_t(\omega_2)\}, \dots, \{X_t(\omega_{2n-1}), X_t(\omega_{2n})\} \right\}.$$

Da n_t^1 eine Bijektion zwischen P_t und Ω definiert, beschreibt S_t eine n -elementige Menge von Paaren von Individuen aus P_t , in denen ein Individuum auch mehrfach auftreten kann. Die Menge S_t kann damit als Teilmenge von $\hat{\mathfrak{A}}^2$ betrachten. Für die Menge aller n -elementigen Teilmengen von $\hat{\mathfrak{A}}^2$ wird das Symbol $(\hat{\mathfrak{A}}^2)^n$ eingeführt. Nun ist es möglich den Selektionsoperator zu definieren:

Definition 2.3 Die fitnessproportionale Selektion O_S zu der Fitnessfunktion f_t ist eine Abbildung

$$\begin{aligned} O_S : \mathbb{P} &\longrightarrow (\hat{\mathfrak{A}}^2)^n && \text{mit} \\ P_t &\mapsto S_t. \end{aligned}$$

Die Menge $O_S(P_t)$ besteht aus ungeordneten Paaren von Individuen aus P_t , wobei die Wahrscheinlichkeit, dass sich ein Individuum $x \in P_t$ in einem beliebigen Paar in der Menge $O_S(P_t)$ befindet, $f_t(x)$ ist.

Rekombination Ein Kreuztausch zwischen zwei Individuen aus $O_S(P_t)$ findet mit einer Wahrscheinlichkeit von p_c statt. Zwei Individuen x_1 und x_2 werden gekreuzt, indem eine Kreuzungsstelle k in ihren Genomen ausgewählt wird und zwei neue Elemente \tilde{x}_1 und \tilde{x}_2 gebildet werden. Eines besteht aus den ersten k Allelen von x_1 und den letzten $(l - k)$ Allelen von x_2 . Das andere Element besteht aus den ersten k Allelen von x_2 und den letzten $(l - k)$ Allelen von x_1 . Abbildung 2.1 veranschaulicht den Vorgang.

Der Kreuztausch agiert auf einer Menge C von Paaren von Individuen:

$$\begin{aligned} C &= \left\{ \{x_1, x_2\}, \{x_3, x_4\}, \dots, \{x_{2n-1}, x_{2n}\} \right\} && x_i \in \mathfrak{A} \quad \forall i \in \{1, \dots, 2n\} \\ C &\in (\hat{\mathfrak{A}}^2)^n. \end{aligned}$$

2 Grundlagen

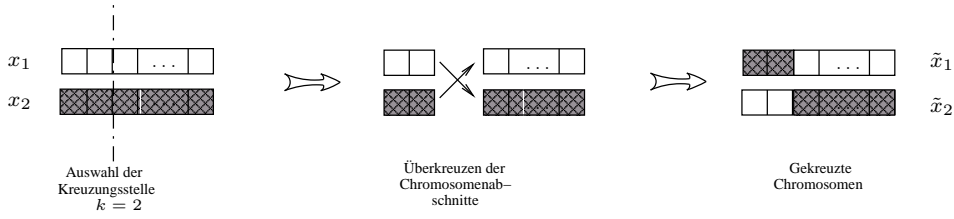


Abbildung 2.1: Beispiel für einen Kreuztausch zwischen x_1 und x_2 mit der Kreuzungsstelle $k = 2$.

Sei $\Omega_1 = \{0, 1, \dots, l-1\}$ eine Ergebnismenge und $f_c : \mathfrak{P}(\Omega_1) \rightarrow \mathbb{R}[0, 1]$ ein Wahrscheinlichkeitsmaß, welches die Kolmogoroffschen Axiome erfüllt und für welches gilt:

$$f_c(\{k\}) =: f_c(k) = \begin{cases} 1 - p_c & : k = 0 \\ p_c \frac{1}{l-1} & : k \neq 0 \end{cases} \quad k \in \Omega_1.$$

Das Tupel $(\Omega_1, \mathfrak{P}(\Omega_1), f_c)$ bildet ebenfalls einen Wahrscheinlichkeitsraum. Eine Zufallsgröße $Y : \Omega_1 \rightarrow \Omega_1 \subset \mathbb{R}$ wird über diesen Wahrscheinlichkeitsraum definiert mit:

$$Y(\omega) = \begin{cases} 0 & : \text{es findet kein Kreuztausch statt} \\ k & : \text{es findet ein Kreuztausch statt und die Kreuzungsstelle ist } k \in \{1, \dots, l-1\} \end{cases}$$

$= k$ mit Wahrscheinlichkeit $f_c(k)$; $\omega, k \in \Omega_1$.

Das Wahrscheinlichkeitsmaß f_c wurde eingeführt, um die Wahrscheinlichkeit, dass zwei Elemente an der Kreuzungsstelle k gekreuzt werden, beschreiben zu können. Die Kreuzungsstelle kann zwischen 1 und $(l-1)$ liegen. Es wurde ein zusätzliches Ereignis $k = 0$ eingeführt, welches dasjenige Ereignis symbolisiert, bei dem überhaupt kein Kreuztausch zweier Elemente stattfindet. Da die Kreuztauschwahrscheinlichkeit p_c beträgt, ist $f_c(0) = 1 - p_c$ die Wahrscheinlichkeit, dass zwei Elemente nicht gekreuzt werden. Findet ein Kreuztausch statt, so ist $f_c(k) = p_c \frac{1}{l-1}$ die Wahrscheinlichkeit, dass die Kreuzungsstelle an der Stelle k gewählt wurde. Damit ist gesichert, dass die Kreuzungsstelle gleichmäßig zwischen 1 und $(l-1)$ verteilt ist.

Ein Zufallsexperiment für Y wird n -mal, also für alle n Paare wiederholt. Das Ereignis im i -ten Versuch ist $Y(\omega_i)$.

Es wird eine Abbildung κ definiert:

$$\kappa : \hat{\mathfrak{X}}^2 \times \Omega_1 \rightarrow \hat{\mathfrak{X}}^2$$

$$\kappa(\{x_1, x_2\}, k) = \begin{cases} \{x_1, x_2\} & : k = 0 \\ \{\tilde{x}_1, \tilde{x}_2\} & : k \neq 0 \end{cases}$$

$$\text{mit} \quad \tilde{x}_1 = (x_1^1, \dots, x_1^k, x_2^{k+1}, \dots, x_2^l)$$

$$\tilde{x}_2 = (x_2^1, \dots, x_2^k, x_1^{k+1}, \dots, x_1^l).$$

Die Menge

$$K' = \{\kappa(\{x_1, x_2\}, Y(\omega_1)), \dots, \kappa(\{x_{2n-1}, x_{2n}\}, Y(\omega_n))\}$$

ist eine Teilmenge von $(\hat{\mathfrak{A}}^2)^n$. Ein Paar kann aus zwei Elementen von P_t bestehen (wenn kein Kreuztausch zwischen ihnen stattfand) oder aus zwei Elementen von \mathfrak{A} , die beim Kreuztausch zwischen zwei Elementen aus P_t entstanden sind. Nun werden die Paare von K' vereinigt, damit man mit einer Menge von Elementen aus \mathfrak{A} arbeiten kann:

$$K = \bigcup_{i=1}^n \kappa(\{x_{2i-1}, x_{2i}\}, Y(\omega_i)).$$

Mit der Hilfe von K kann der Kreuztauschoperator definiert werden:

Definition 2.4 Der 1-Punkt Kreuztauschoperator O_K ist eine Abbildung

$$\begin{aligned} O_K : (\hat{\mathfrak{A}}^2)^n &\longrightarrow \mathbb{P} && \text{mit} \\ C &\mapsto K. \end{aligned}$$

Wird O_K auf eine Menge C angewendet, so werden die zwei Elemente je eines Paares mit der Wahrscheinlichkeit p_c gekreuzt. Findet ein Kreuztausch zwischen zwei Elementen statt, so werden sie an einer gleichmäßig im Genom verteilten Stelle gekreuzt. Das Resultat des 1-Punkt Kreuztauschoperator ist eine Menge, die aus gekreuzten oder ungekreuzten Elementen von P_t besteht.

Mutation Nun soll der Mutationsoperator formuliert werden. Dieser wird allgemein für eine $2n$ -elementige Menge $M \in \mathbb{P}$ mit Elementen aus \mathfrak{A} definiert. Jedes Allel eines Genoms wird mit der Mutationswahrscheinlichkeit p_m mutiert. Findet eine Mutation statt, so wird eine „1“ durch eine „0“ ersetzt und umgekehrt.

Zu Beginn werden die Elemente von M mit Hilfe der Bijektion $n_2 : M \longrightarrow \Omega$ nummeriert. Es wird der Ergebnisraum $\Omega_2 = \{0, 1\}$ und das Wahrscheinlichkeitsmaß $f_m : \mathfrak{P}(\Omega_2) \longrightarrow \mathbb{R}$ eingeführt. Für f_m soll gelten:

$$f_m(\{q\}) =: f_m(q) = \begin{cases} p_m & : q = 1 \\ 1 - p_m & : q = 0. \end{cases} \quad q \in \Omega_2$$

Damit ist das Tupel $(\Omega_2, \mathfrak{P}(\Omega_2), f_m)$ ein Wahrscheinlichkeitsraum über dem die Zufallsgröße $Z : \Omega_2 \longrightarrow \Omega_2 \subset \mathbb{R}$ definiert wird:

$$\begin{aligned} Z(\omega) &= \begin{cases} 0 & : \text{es findet keine Mutation statt} \\ 1 & : \text{es findet eine Mutation statt} \end{cases} \\ &= \begin{cases} 0 & : \text{mit der Wahrscheinlichkeit } f_m(0) \\ 1 & : \text{mit der Wahrscheinlichkeit } f_m(1) \end{cases} \quad \omega \in \Omega_2. \end{aligned}$$

2 Grundlagen

Die Zufallsgröße Z beschreibt das Ereignis, ob eine Mutation an einem Allel stattfindet oder nicht. Das Zufallsexperiment für Z wird für jedes Element von M genau l -mal durchgeführt. Das Ereignis des j -ten Versuches für das i -te Element ist $Z^i(\omega_j)$.

Es wird eine Abbildung m_j definiert, die einem Element dasjenige Genom zuordnet, welches bei Stattfinden der Mutation dieses Elementes am j -ten Allel entsteht:

$$m_j : \mathfrak{A} \times \Omega_2 \longrightarrow \mathfrak{A}$$

$$m_j(x, q) = \begin{cases} (x^1, \dots, x^j, \dots, x^l) & : q = 0 \\ (x^1, \dots, 1 - x^j, \dots, x^l) & : q = 1 \end{cases} .$$

Nun wird die Abbildung m eingeführt, welche einem Element dasjenige Element zuordnet, welches nach l Zufallsexperimenten für Z entsteht:

$$m : \mathfrak{A} \longrightarrow \mathfrak{A}$$

$$m(x_i) = m_l \left(\dots m_2 \left(m_1 \left(x_i, Z^i(\omega_1) \right), Z^i(\omega_2) \right) \dots, Z^i(\omega_l) \right).$$

Damit kann nun die Menge M' als Menge der Elemente aus \mathfrak{A} , die nach dem Stattfinden oder Nicht-Stattfinden von Mutationen an den Allelen der Genome von M entstanden ist, beschrieben werden:

$$M' = \{m(x_1), m(x_2), \dots, m(x_{2n})\}.$$

Mit der Hilfe von M' kann nun der Mutationsoperator definiert werden:

Definition 2.5 *Der Operator der bitweisen Mutation O_M ist eine Abbildung*

$$O_M : \mathbb{P} \longrightarrow \mathbb{P}$$

$$M \mapsto M'.$$

Das genetische Verfahren Nach der Definition der Operatoren F , O_S , O_K und O_M wird nun das genetische Verfahren konkretisiert: Ist eine Anfangspopulation P_0 gegeben so erzeugt das genetische Verfahren eine Folge von Populationen $\{P_t\}_{t \geq 1}$ mit folgender Iterationsvorschrift:

$$P_{t+1} = O_M(O_K(O_S(P_t))) \quad t \in \mathbb{N}_0.$$

Zu Beginn einer Iteration werden also die Individuen von P_t selektiert und auf diese der Kreuztauschoperator angewendet. Auf die resultierenden Elemente wird der Mutationsoperator angewendet.

Die Werte für die Wahrscheinlichkeiten p_m und p_c werden bei dem einfachen genetischen Algorithmus konstant gesetzt, sind also nicht von der Generation t abhängig. In der Praxis werden häufig p_c nahe 1 und p_m nahe 0 gewählt, da für diese Belegung gute Ergebnisse erzielt wurden.

2.2 Einführung in die Schematheorie

In diesem Abschnitt wird eine kurze Einführung in die Schematheorie gegeben. Konzept und Bezeichnungen sind von [Hol75] übernommen. Die Schematheorie beschreibt die Konstruktion von Schemen. Diese sind von der verwendeten Kodierung abhängig. Zunächst wird die Konstruktion von allgemeinen Schemen beschrieben und im Anschluss daran die Spezialisierung auf binäre Schemen für Genetische Algorithmen.

Das Ziel der Schematheorie ist die Suche nach einer Möglichkeit, die Elemente des Suchraumes miteinander zu vergleichen. Dabei wird vom direkten Vergleich der Zielfunktionswerte abgesehen. Mittlerweile gibt es mehr Beweggründe für die Schemennotation. Darauf wird im Kapitel 6 eingegangen.

Um bestimmte Eigenschaften der Elemente zu quantifizieren, werden Detektoren definiert.

Definition 2.6 *Ein Detektor δ ist eine Abbildung des Suchraumes \mathfrak{A} in den Wertevorrat V .*

$$\delta : \mathfrak{A} \longrightarrow V.$$

Die Menge von Detektoren wird mit Δ bezeichnet:

$$\Delta = \{\delta_i : \mathfrak{A} \longrightarrow V_i, i = 1, \dots, m\} \quad m \in \mathbb{N}.$$

Dabei ist V_i der Wertevorrat des i -ten Detektors und m die Anzahl der Detektoren.

Mit Hilfe der Detektoren erhält jedes Element $x \in \mathfrak{A}$ eine Darstellung mit dem geordneten Tupel $(\delta_1(x), \delta_2(x), \dots, \delta_m(x))$.

Nicht immer ist es von Interesse, den Wert eines bestimmten Detektors zu kennen. Für diesen Fall wird das Symbol „*“ eingeführt. Damit beschreibt das Tupel $(s_1, *, \dots, *, s_l)$ die Menge aller Elemente $x \in \mathfrak{A}$, für die gilt:

$$\begin{aligned} \delta_1(x) &= s_1 & s_1 &\in V_1 \\ \delta_l(x) &= s_l & s_l &\in V_l. \end{aligned}$$

Die Werte aller anderen Detektoren $\delta_2(x), \dots, \delta_{l-1}(x)$ sind hierbei nicht von Belang.

Definition 2.7 *Ein Schema ist ein m -Tupel aus der Menge Ξ*

$$\Xi = \prod_{i=1}^m \{V_i \cup \{*\}\}.$$

2 Grundlagen

Ist $H = (s_1, s_2, \dots, s_m)$, so beschreibt H die Menge aller Elemente aus \mathfrak{A} , für welche gilt:

$$\delta_i(x) = s_i \quad \text{falls } s_i \neq * \quad \forall i = 1, \dots, m.$$

Das Schema wird mit der Menge, die es beschreibt identifiziert.

Jedes Schema unterteilt den Suchraum \mathfrak{A} in zwei Teilmengen, in H und $\mathfrak{A} \setminus H$. Ein Element $x \in \mathfrak{A}$ heißt *Vertreter* eines Schemas H , falls $x \in H$. Die Anzahl der Vertreter eines Schemas H in der Population P_t wird mit $m(H, t) = |P_t \cap H|$ bezeichnet. Eine Position $k \in \{1, \dots, m\}$ eines Schemas $H = (s_1, s_2, \dots, s_m)$ heißt *fixiert*, wenn $s_k \neq *$.

Definition 2.8 Die *Ordnung* $o(H)$ eines Schemas $H \in \Xi$ ist die Anzahl der fixierten Stellen von H .

Definition 2.9 Die *Länge* $\delta(H)$ eines Schemas $H \in \Xi$ ist die Differenz zwischen der ersten und der letzten fixierten Stelle von H .

Definition 2.10 Die *durchschnittliche Fitness* $F_t(H)$ eines Schemas in der Population P_t ist

$$F_t(H) = \frac{1}{m(H, t)} \sum_{x \in P_t \cap H} F(x).$$

Definition 2.11 Die *relative Fitness* $f_t(H)$ eines Schemas H ist die Summe der relativen Fitnesswerte der Vertreter des Schemas, die sich in der Population P_t befinden:

$$f_t(H) = \sum_{x \in P_t \cap H} f_t(x).$$

Beginnend bei der ersten fixierten Position eines Schemas H werden die Zeichenzwischenräume mit h_1 bis $h_{\delta(H)}$ bezeichnet.

Anpassung an den einfachen genetischen Algorithmus Für den einfachen genetischen Algorithmus werden die Detektoren folgendermaßen spezialisiert: Die Anzahl der Detektoren ist identisch mit der Genomlänge l und der Wertevorrat jedes Detektors ist die Menge der möglichen Allele an einem Gen:

$$V_i = \{0, 1\} \quad i = 1, \dots, l.$$

Weiterhin wird festgelegt, dass der i -te Detektor das Allel am i -ten Gen wiedergeben soll:

2.3 Zusammenfassung

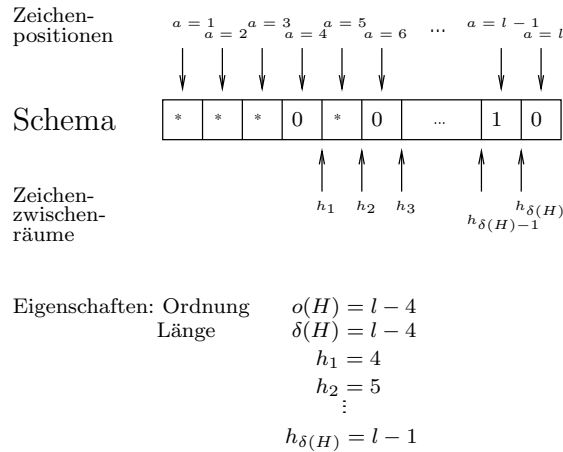


Abbildung 2.2: Bezeichnungen an einem binären Schema.

$$\delta_i(x) = \begin{cases} 0 & : \text{ das Allel am } i\text{-ten Gen von } x \text{ ist eine } 0 \\ 1 & : \text{ das Allel am } i\text{-ten Gen von } x \text{ ist eine } 1 \end{cases} \quad x \in \mathfrak{A}.$$

Abbildung 2.2 veranschaulicht die Bezeichnungen an einem Schema für den einfachen genetischen Algorithmus.

2.3 Zusammenfassung

In diesem Kapitel wurde das genetische Verfahren konstruiert, das dem einfachen genetischen Algorithmus zugrundeliegt. Dazu wurden Abbildungen für die Fitnesszuweisung, die Selektion, die Rekombination und die Mutation eingeführt.

Weiterhin wurde gezeigt, wie ein Schema mit Hilfe von Detektoren konstruiert werden kann. Ein Detektor wird dabei als Hilfsmittel benutzt, mit dem gewisse Eigenschaften der Elemente von \mathfrak{A} quantifiziert werden können. Ein Schema vereinigt die Elemente zu einer Menge, die dieselben Ausprägungen bestimmter Eigenschaften besitzen. Somit ermöglichen Schemen eine Unterteilung des Suchraumes. Mit dieser Unterteilung ist es wiederum möglich, die Elemente untereinander bezüglich ihrer Ausprägung bestimmter Eigenschaften zu vergleichen. Die Einführung in die Schematheorie in diesem Kapitel ist an [Hol75] angelehnt.

2 Grundlagen

3 Hollands Schemensatz und die Evolutionsgleichungen

In diesem Kapitel soll Hollands Schemensatz vorgestellt, hergeleitet und anschließend interpretiert werden. Danach wird die Ungleichung, die im Schemensatz gegeben ist, durch eine Gleichung verbessert. Es wird gezeigt, dass dieser modifizierte Schemensatz mit dem von Stephens übereinstimmt. Zum Abschluss wird der Schemensatz auf beliebige genetische Operatoren verallgemeinert und bezüglich der Homogenisierung der Population analysiert.

Es wird ein einfacher genetischer Algorithmus mit der Population P_t betrachtet.

3.1 Hollands Schemensatz

Satz 3.1 (Hollands Schemensatz) *In einem einfachen genetischen Algorithmus gilt für die erwartete Anzahl von Vertretern eines Schemas H in der Folgegeneration [Hol75]:*

$$E(m(H, t + 1)) \geq m(H, t) \frac{F_t(H)}{\bar{F}_t} \left(1 - p_c \frac{\delta(H)}{l-1} (1 - f_t(H)) \right) (1 - p_m)^{o(H)} \quad (3.1)$$

Im Folgenden soll der Satz 3.1 hergeleitet werden. Damit ein Schema einen Vertreter in der Folgepopulation besitzt, muss entweder ein Vertreter von H die genetischen Operatoren unverändert überstehen oder durch die Rekombination und Mutation konstruiert werden. In Hollands Schemensatz wird nur der erste Fall betrachtet: Ein Vertreter von H wird selektiert und die fixierten Stellen von H überstehen Rekombination und Mutation unverändert. Die Wahrscheinlichkeit eines Schemas selektiert zu werden ist die Summe der relativen Fitnesswerte seiner Vertreter in P_t , also $f_t(H)$. Es werden $2n$ Individuen für den Kreuztausch ausgewählt und somit ist die erwartete Anzahl der ausgewählten Individuen, die Vertreter von H sind:

3 Hollands Schemensatz und die Evolutionsgleichungen

$$\begin{aligned}
2nf_t(H) &= 2n \sum_{x \in H \cap P_t} f_t(x) \\
&= 2n \sum_{x \in H \cap P_t} \frac{F(x)}{\sum_{y \in P_t} F(y)} \quad \text{wenn } \exists x \in P_t : F(x) > 0 \\
&= 2n \frac{\sum_{x \in H \cap P_t} F(x)}{\sum_{y \in P_t} F(y)} \frac{m(H, t)}{m(H, t)} \\
&= m(H, t) \frac{\frac{1}{m(H, t)} \sum_{x \in H \cap P_t} F(x)}{\frac{1}{2n} \sum_{y \in P_t} F(y)} \\
&= m(H, t) \frac{F_t(H)}{\bar{F}_t}. \tag{3.2}
\end{aligned}$$

Tritt der Fall $F(x) = 0 \forall x \in P_t$ ein, so gilt:

$$\begin{aligned}
2nf_t(H) &= 2n \sum_{x \in H \cap P_t} f_t(x) \\
&= 2n \sum \frac{1}{2n} \\
&= 2nm(H, t) \frac{1}{2n} \\
&= m(H, t).
\end{aligned}$$

Im Folgenden wird dieser Fall jedoch vernachlässigt, da seine gesonderte Behandlung keine neuen Einsichten liefert, sondern nur die Untersuchungen erschwert.

Werden zwei Individuen gekreuzt, wovon wenigstens eines ein Vertreter von H ist, so interessiert, wie wahrscheinlich es ist, dass von den beiden aus der Kreuzung resultierenden Elementen wenigstens einer ein Vertreter von H ist. Dazu wird das komplementäre Ereignis betrachtet: Keines der aus dem Kreuztausch resultierenden Elementen ist ein Vertreter von H . Dieses Ereignis kann eintreten, wenn ein Vertreter von H mit einem Individuum $x \notin H$ gekreuzt wird und die Kreuzungsstelle zwischen die fixierten Positionen von H fällt. Die Wahrscheinlichkeit hierfür ist:

$$\begin{aligned}
P(\text{Kreuzung zw. } x_1 \text{ und } x_2 \text{ an } k, \{x_1, x_2\} \cap (\mathfrak{A} \setminus H) \neq \emptyset \text{ und} \\
k \in \{h_1, \dots, h_{\delta(H)}\}) &= \delta(H) f_c(k) (1 - f_t(H)) \\
&= p_c \frac{\delta(H)}{l-1} (1 - f_t(H)).
\end{aligned}$$

Damit ist die Wahrscheinlichkeit, dass bei der Kreuzung zweier Individuen an der Kreuzungsstelle k wenigstens ein Vertreter von H resultiert:

$$P(\{\tilde{x}_1, \tilde{x}_2\} \cap H \neq \emptyset) \geq 1 - p_c \frac{\delta(H)}{l-1} (1 - f_t(H)). \quad (3.3)$$

Man erhält eine untere Schranke in (3.3), da die tatsächliche Wahrscheinlichkeit für Schemen nach einer Kreuzung Vertreter zu haben größer ist als der in (3.3) gegebene Term. Eine Kreuzung kann auch zwischen einem Vertreter von H und einem Individuum stattfinden, welches Vertreter eines Teilschemas ist, das bis zur Kreuzungsstelle mit H übereinstimmt. In diesem Fall würden auch Vertreter von H aus dem Kreuztausch resultieren, doch dieser Fall wird in (3.3) nicht beachtet.

Findet eine Mutation an einem Element statt, welches ein Vertreter von H ist, so ist das aus der Mutation resultierende Element ein Vertreter von H , wenn keine Mutation an den fixierten Stellen von H stattfand. Die Wahrscheinlichkeit, dass $o(H)$ -mal keine Mutation in dem Genom des Elementes stattfindet ist

$$\begin{aligned} P(o(H)\text{-mal keine Mutation}) &= P\left(\sum_{i=1}^l Z(\omega_i) \leq l - o(H)\right) \\ &= \binom{l}{o(H)} (1 - p_m)^{o(H)}, \end{aligned}$$

da es $\binom{l}{o(H)}$ Möglichkeiten gibt, $o(H)$ Positionen in einem Genom der Länge l auszuwählen. Der Ausdruck

$$\sum_{i=1}^l Z(\omega_i) \leq l - o(H)$$

bedeutet, dass bei der Summation der Ereignisse, die beim l -maligen Durchführen des Zufallsexperimentes für Z eintreten, die Summe nicht größer als $l - o(H)$ sein darf, es muss also wenigstens $o(H)$ -mal $Z(\omega_i) = 0$ eingetreten sein. In dem Genom gibt es nur eine Auswahl von $o(H)$ Positionen, so dass die fixierten Positionen von H ausgewählt werden. Folglich ist die Wahrscheinlichkeit, dass keine Mutation an den fixierten Positionen auftritt:

$$\begin{aligned} P(m(x) \in H) &= \binom{l}{o(H)} (1 - p_m)^{o(H)} \frac{1}{\binom{l}{o(H)}} \\ &= (1 - p_m)^{o(H)}. \end{aligned} \quad (3.4)$$

Wie bereits erwähnt wurde, kann ein Schema H in der Folgepopulation Vertreter besitzen, wenn Vertreter von H bei der Selektion ausgewählt werden und auch nach dem Anwenden des Kreuztausch- und des Mutationsoperators noch Vertreter von H sind. Eine untere Schranke für die

3 Hollands Schemensatz und die Evolutionsgleichungen

Wahrscheinlichkeit, dass ein Vertreter von H aus dem Kreuztausch eines Vertreters von H mit einem anderen Element resultiert, ist mit (3.3) gegeben. Die Wahrscheinlichkeit, dass aus der Mutation eines Vertreters von H ein Element resultiert, welches ebenfalls in H liegt, ist mit (3.4) gegeben. Gleichung (3.2) beschreibt die erwartete Anzahl von Vertretern von H , die zum Kreuztausch ausgewählt werden. Multipliziert man diese drei Terme miteinander, so erhält man eine untere Schranke für den Erwartungswert der Anzahl an Vertretern eines Schemas H in der Folgepopulation P_{t+1} :

$$\begin{aligned} E(m(H, t+1)) &\geq m(H, t) \frac{F_t(H)}{\bar{F}_t} \left(1 - p_c \frac{\delta(H)}{l-1} (1 - f_t(H))\right) (1 - p_m)^{o(H)} \\ &= 2n f_t(H) \left(1 - p_c \frac{\delta(H)}{l-1} (1 - f_t(H))\right) (1 - p_m)^{o(H)} \end{aligned} \quad (3.5)$$

Somit ist Satz 3.1 bewiesen.

Interpretation von Hollands Schemensatz Hollands Schemensatz 3.1 schätzt die Anzahl der in der Folgegeneration zu erwartenden Vertreter eines Schemas nach unten ab. Eine Folgerung aus dem Schemensatz wird jedoch an einer schwächeren Form der in (3.1) gegebenen unteren Schranke ersichtlich. Diese schwächere Form ist gegeben durch:

$$\begin{aligned} E(m(H, t+1)) &\geq m(H, t) \frac{F_t(H)}{\bar{F}_t} \left(1 - p_c \frac{\delta(H)}{l-1} (1 - f_t(H))\right) (1 - p_m)^{o(H)} \\ &\geq m(H, t) \frac{F_t(H)}{\bar{F}_t} \left(1 - p_c \frac{\delta(H)}{l-1}\right) (1 - p_m)^{o(H)}, \end{aligned} \quad (3.6)$$

da $(1 - f_t(H)) \geq 0$.

In der Praxis werden p_c nahe 1 und p_m nahe 0 gewählt:

$$p_m \approx 0, \quad p_c \approx 1.$$

Somit hängt die Vertreteranzahl eines Schemas H in der Folgepopulation im Wesentlichen von dem Verhältnis seiner durchschnittlichen Fitness zu der durchschnittlichen Populationsfitness $\frac{F_t(H)}{\bar{F}_t}$, seiner Länge $\delta(H)$ und seiner Ordnung $o(H)$ ab. Ist $F_t(H)$ größer als \bar{F}_t , $\frac{\delta(H)}{l-1}$ und $o(H)$ klein, dann wird die Anzahl der Vertreter von H vergrößert. Ist $F_t(H)$ kleiner oder gleich \bar{F}_t , so wird die Anzahl dezimiert.

Bleibt ein kleines ($o(H)$ gering) und kompaktes ($\frac{\delta(H)}{l-1}$ gering) Schema über mehrere Generationen mit einem bestimmten Betrag $c\bar{F}_t$ über der durchschnittlichen Populationsfitness \bar{F}_t ($c > 0$):

$$F_t(H) = (1 + c)\bar{F}_t,$$

so kann man zeigen, dass es in den Folgepopulation eine exponentiell steigende Anzahl von Vertretern besitzt. Ist das Schema H klein und kompakt so kann die destruktive Wirkung von Kreuztausch und Mutation vernachlässigt werden:

$$1 - p_c \frac{\delta(H)}{l-1} \approx 1 \qquad (1 - p_m)^{o(H)} \approx 1.$$

Und sofern $m(H, 0) \neq 0$ sieht man:

$$\begin{aligned} E(m(H, t+1)) &\geq m(H, t) \frac{F_t(H)}{\bar{F}_t} \left(1 - p_c \frac{\delta(H)}{l-1}\right) (1 - p_m)^{o(H)} \\ &\approx m(H, t) \frac{F_t(H)}{\bar{F}_t} \\ &= m(H, t)(1 + c) \\ &= m(H, t-1)(1 + c)^2 \\ &\vdots \\ &= m(H, 0)(1 + c)^t. \end{aligned}$$

Kleine, kompakte und überdurchschnittlich fit bleibende Schemen erhalten also eine exponentiell steigende Anzahl an Vertretern in der Population, da die untere Schranke für die erwartete Anzahl an Vertretern dieser Schemen mit einer Rate von $(1 + c)$ exponentiell wächst. In einer endlichen Population kann natürlich kein Schema fortlaufend überdurchschnittlich fit bleiben, da es mit steigender Vertreteranzahl auch die durchschnittliche Populationsfitness \bar{F}_t erhöht. Aus diesem Grund und wegen der endlichen Populationsgröße kann somit auch nicht die Vertreteranzahl eines Schemas fortlaufend exponentiell ansteigen. Doch die Hauptaussage des Schemensatzes bleibt bestehen: In einem einfachen genetischen Algorithmus erhalten überdurchschnittlich fitte Schemen mit bestimmten Eigenschaften (klein und kompakt) über einige Generationen hinweg eine exponentiell steigende Vertreteranzahl in der Population. Da sich die durchschnittliche Fitness eines Schemas aus den Fitnesswerten der Individuen zusammensetzt, die Vertreter des Schemas sind, deutet ein überdurchschnittlich fittes Schema auch immer auf überdurchschnittlich fitte Vertreter in P_t hin. Steigt also die Vertreteranzahl eines Schemas, welches überdurchschnittlich fit ist, steigt auch die Anzahl an überdurchschnittlich fitten Individuen. Somit kann der Schemensatz 3.1 eine Erklärung für das schnelle Finden einer guten Lösung durch einfache genetische Algorithmen bieten.

3.2 Modifikation von Hollands Schemensatz

In Hollands Schemensatz 3.1 wird nur der destruktive Einfluss des Kreuztauses und der Mutation auf die Vertreteranzahl eines Schemas in der Folgepopulation betrachtet. Dadurch bietet der Schemensatz nur eine untere Schranke für eben jene Vertreteranzahl. Da jedoch für die Analyse des einfachen genetischen Algorithmus eine präzisere Angabe wünschenswert ist, wird für die erwartete Vertreteranzahl eines Schemas in der Folgepopulation eine exakte Formel hergeleitet.

Bei dem einfachen genetischen Algorithmus können durch folgende Ursachen Vertreter eines Schemas H in der Folgepopulation vorhanden sein:

- (1a) Aus der Kreuzung zweier Individuen, wovon wenigstens eines in H liegt, resultiert wenigstens ein Vertreter von H .
- (1b) Bei der Kreuzung zweier Individuen, welche nicht Vertreter von H sind, resultiert ein Element, das in H liegt.
- (2) Aus der Mutation von einem Vertreter von H resultiert ein Element, welches ebenfalls in H liegt.
- (3) Aus der Mutation von einem Element, welches kein Vertreter von H ist, resultiert ein Vertreter von H .

Diese vier Szenarien sollen im Folgenden betrachtet und hinsichtlich ihres Einflusses auf die Vertreteranzahl eines Schemas in der Folgepopulation analysiert werden. Es wurde nur (2) und teilweise (1a) in Hollands Schemensatz 3.1 betrachtet.

Zunächst wird der Fall betrachtet, dass einige Elemente nach dem Anwenden des Kreuztauschoperators Vertreter von H sind:

$$O_K(S_t) \cap H \neq \emptyset.$$

(1a) Es wird zuerst das Ereignis untersucht, bei dem wenigstens ein Vertreter von H an der Kreuzung zwischen zwei Individuen teilnimmt. Bei dem Kreuztausch zwischen zwei Individuen wird eine Kreuzungsstelle ausgewählt. Es werden zwei Fälle unterschieden. Die Kreuzungsstelle $k \in \{1, \dots, l-1\}$ kann außerhalb des Schemas H oder zwischen die fixierten Positionen des Schemas fallen.

Die Wahrscheinlichkeit, dass zwischen zwei Individuen ein Kreuztausch stattfindet und die Kreuzungsstelle zwischen die fixierten Positionen fällt, ist:

$$P(k \in \{h_1, \dots, h_{\delta(H)}\}) = p_c \frac{\delta(H)}{l-1}. \quad (3.7)$$

Dann ist die Wahrscheinlichkeit, dass k außerhalb der fixierten Stellen von H liegt:

$$P(k \notin \{h_1, \dots, h_{\delta(H)}\}) = 1 - p_c \frac{\delta(H)}{l-1}.$$

Bei einer Kreuzung zweier Individuen, bei der wenigstens ein Vertreter von H teilnimmt, können entweder beide oder nur ein Individuum Vertreter von H sein. Die Auswahl eines Vertreters von H und eines Individuums, welches kein Vertreter von H ist, kann dabei auf zwei Arten erfolgen. Der Vertreter von H kann als erster oder als zweiter Kreuzungspartner gewählt werden. Sind beide Individuen Vertreter von H , so gibt es nur eine Auswahlmöglichkeit. Damit ist die Wahrscheinlichkeit, dass wenigstens eines von zwei zu kreuzenden Individuen ein Vertreter von H ist:

$$P(\{x_1, x_2\} \cap H \neq \emptyset) = f_t(H)^2 + 2f_t(H)(1 - f_t(H)).$$

Die Anzahl der Vertreter von H nach einem Kreuztausch zwischen zwei Individuen ist abhängig davon, wieviele Vertreter von H , ob zwei oder nur einer, am Kreuztausch beteiligt waren. Werden zwei Vertreter von H miteinander gekreuzt, liegen beide entstehende Elemente in H . Ist jedoch nur ein Individuum Vertreter von H , so liegt auch nur eines von zwei resultierenden Elementen in H . Werden zwei Individuen gekreuzt, wobei wenigstens eines Element von H ist, so ist die zu erwartende Anzahl von Vertretern von H demnach:

$$\begin{aligned} E \left[\{x_1, x_2\} \cap H \neq \emptyset, \kappa(\{x_1, x_2\}, k) \cap H \neq \emptyset, k \notin \{h_1, \dots, h_{\delta(H)}\} \right] &= \\ &= \left(1 - p_c \frac{\delta(H)}{l-1} \right) (2f_t(H)^2 + 1 \cdot 2f_t(H)(1 - f_t(H))) \\ &= 2f_t(H) \left(1 - p_c \frac{\delta(H)}{l-1} \right). \end{aligned} \quad (3.8)$$

Damit wurde Szenario (1a) vollständig untersucht.

(1b) Nun wird das Ereignis betrachtet, bei dem die Kreuzungsstelle zwischen die fixierten Positionen von H fällt. Die Wahrscheinlichkeit für dieses Ereignis wurde bereits erwähnt (3.7):

$$P(k \in \{h_1, \dots, h_{\delta(H)}\}) = p_c \frac{\delta(H)}{l-1}.$$

Nach dem Kreuztausch zweier Individuen, bei dem wenigstens eines der resultierenden Elemente in H liegt und die Kreuzungsstelle zwischen die fixierten Positionen fällt, können ein oder zwei Elemente in H liegen. Das

3 Hollands Schemensatz und die Evolutionsgleichungen

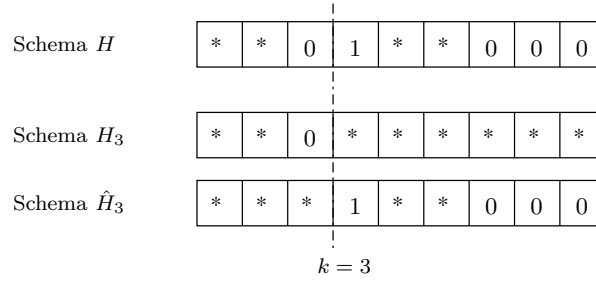


Abbildung 3.1: Ein Beispiel für die Schemen H , H_k und \hat{H}_k mit $k = 3$.

ist davon abhängig, ob zwei Vertreter von H miteinander gekreuzt werden oder ob ein Vertreter von H mit einem Individuum gekreuzt wird, welches entweder bis zur Kreuzungsstelle oder von der Kreuzungsstelle an den fixierten Positionen von H mit dessen Allelen übereinstimmt. Für diesen Fall wird folgende Notation eingeführt.

Definition 3.1 Das Schema $H_k \in \Xi$ ist ein Schema, dessen fixierte Positionen und dessen Allele an diesen fixierten Positionen mit denen von H bis zur Zeichenposition $k \in \{h_1, \dots, h_{\delta(H)}\}$ übereinstimmen. Abgesehen von diesen erwähnten fixierten Positionen sind keine weiteren Positionen von H_k fixiert. Ist $H = (s_1, \dots, s_k, \dots, s_l)$, dann ist also $H_k = (s_1, \dots, s_k, *, \dots, *)$.

Definition 3.2 Das Schema $\hat{H}_k \in \Xi$ ist ein Schema, dessen fixierte Positionen und dessen Allele an diesen fixierten Positionen mit denen von H von der Zeichenposition $k \in \{h_1, \dots, h_{\delta(H)}\}$ an übereinstimmen. Abgesehen von diesen erwähnten fixierten Positionen sind keine weiteren Positionen von \hat{H}_k fixiert. Ist $H = (s_1, \dots, s_k, \dots, s_l)$, dann ist also $\hat{H}_k = (*, \dots, *, s_{k+1}, \dots, s_l)$.

Abbildung 3.1 zeigt ein Beispiel für H , H_k und \hat{H}_k . Es sind folgende Beziehungen erkennbar:

Hilfssatz 3.2 Für H , H_k und \hat{H}_k gelten folgende Beziehungen:

- a) $H \subseteq H_k$
- b) $H \subseteq \hat{H}_k$
- c) $H_k \cap \hat{H}_k = H$.

Beweis: Sei $H = (s_1, \dots, s_k, \dots, s_l)$, $H_k = (s_1, \dots, s_k, *, \dots, *)$ und $\hat{H}_k = (*, \dots, *, s_{k+1}, \dots, s_l)$. Sei $x \in H$, also ist der Detektor $\delta_i(x) = s_i$ für $s_i \neq *$, $i = 1, \dots, l$.

Zu a) Damit ist x auch Element von H_k . Es folgt $H \subseteq H_k$.

Zu b) Um b) zu beweisen geht man entsprechend a) vor, nur dass H_k

3.2 Modifikation von Hollands Schemensatz

durch \hat{H}_k ersetzt wird.

zu c) Sei $x \in H_k$ und $x \in \hat{H}_k$, also gilt für die Detektoren $\delta_i(x) = s_i$ für $i = 1, \dots, k$ und für $i = k + 1, \dots, l$.

$\Rightarrow x \in H$

$\Rightarrow H = H_k \cap \hat{H}_k$. □

Folgerung 3.3 Für die relativen Fitnesswerte von H, H_k und \hat{H}_k gelten folgende Beziehungen:

$$a_1) f_t(H_k) \geq f_t(H) \qquad a_2) f_t(H_k \setminus H) = f_t(H_k) - f_t(H)$$

$$b_1) f_t(\hat{H}_k) \geq f_t(H) \qquad b_2) f_t(\hat{H}_k \setminus H) = f_t(\hat{H}_k) - f_t(H)$$

$$c) f_t(H_k \cup \hat{H}_k) = f_t(H_k) + f_t(\hat{H}_k) - f_t(H).$$

Beweis zu $a_1)$ Nach dem Hilfssatz 3.2 ist H eine Teilmenge von H_k . Damit besitzt H_k wenigstens so viele Vertreter, wie H . Dies gilt auch für die Anzahl der Vertreter in der Population. Da sich die relative Fitness eines Schemas aus der Summe der relativen Fitnesswerte seiner Vertreter in der Population errechnet, ist der relative Fitnesswert von H_k notwendigerweise mindestens so groß, wie der von H . Der Beweis wird jedoch auch formell durchgeführt, damit $a_2)$ leichter zu beweisen ist.

$$\begin{aligned} f_t(H_k) &= \sum_{x \in P_t \cap H_k} f_t(x) \\ &= \sum_{x \in P_t \cap ((H_k \setminus H) \cup H)} f_t(x) \quad \text{mit Hilfssatz 3.2 a)} \\ &= \sum_{x \in (P_t \cap (H_k \setminus H)) \cup (P_t \cap H)} f_t(x) \\ &= \sum_{x \in P_t \cap (H_k \setminus H)} f_t(x) + \sum_{x \in P_t \cap H} f_t(x) \\ &\geq \sum_{x \in P_t \cap H} f_t(x) \\ &= f_t(H) \end{aligned}$$

zu $a_2)$

$$\begin{aligned} f_t(H_k) &= \sum_{x \in P_t \cap (H_k \setminus H)} f_t(x) + \sum_{x \in P_t \cap H} f_t(x) \\ &= f_t(H_k \setminus H) + f_t(H) \end{aligned}$$

Also ist $f_t(H_k \setminus H) = f_t(H_k) - f_t(H)$

zu $b_1)$ und $b_2)$ Man verfährt genauso wie in $a_1)$ und $a_2)$ mit der Ausnahme,

3 Hollands Schemensatz und die Evolutionsgleichungen

dass H_k durch \hat{H}_k zu ersetzen ist.
zu c)

$$\begin{aligned}
f_t(H_k \cup \hat{H}_k) &= \sum_{x \in P_t \cap (H_k \cup \hat{H}_k)} f_t(x) \\
&= \sum_{x \in (P_t \cap H_k) \cup (P_t \cap \hat{H}_k)} f_t(x) \\
&= \sum_{x \in (P_t \cap H_k)} f_t(x) + \sum_{x \in (P_t \cap H)} f_t(x) - \sum_{x \in (P_t \cap H_k) \cap (P_t \cap \hat{H}_k)} f_t(x) \\
&= f_t(H_k) + f_t(\hat{H}_k) - \sum_{x \in (P_t \cap H)} f_t(x) \quad \text{mit Satz 3.2 c)} \\
&= f_t(H_k) + f_t(\hat{H}_k) - f_t(H).
\end{aligned}$$

□

Es gibt vier Kombinationen, in denen Vertreter von H , H_k und \hat{H}_k miteinander gekreuzt werden können, so dass wenigstens ein aus dem Kreuztausch hervorgehendes Element Vertreter von H ist. Folgende Tabelle gibt einen Überblick über diese Kombinationen, die Anzahl der Möglichkeiten, diese Kombinationen auszuwählen und die Anzahl der resultierenden Elemente, die bei diesen Kombinationen in H liegen:

Kombination	Anzahl der Arten, die diese Kombination auszuwählen	Resultierende Anzahl von Vertretern von H
H und H	1	2
H und $H_k \setminus H$	2	1
H und $\hat{H}_k \setminus H$	2	1
$H_k \setminus H$ und $\hat{H}_k \setminus H$	2	1

Die Wahrscheinlichkeit, dass Vertreter von H , H_k oder \hat{H}_k in einer derartigen Konstellation gekreuzt werden ist demnach

$$\begin{aligned}
&P(\kappa(\{x_1, x_2\}, k_0) \cap H \neq \emptyset, k_0 \in \{h_1, \dots, h_{\delta(H)}\}) = \\
&= P(k_0 \in \{h_1, \dots, h_{\delta(H)}\}) P(k_0 = k) \left(P(x_1 \in H, x_2 \in H) + \right. \\
&\quad P(x_1 \in H, x_2 \in H_k) + P(x_1 \in H, x_2 \in \hat{H}_k) + \\
&\quad \left. P(x_1 \in H_k, x_2 \in \hat{H}_k) \right) \\
&= p_c \frac{\delta(H)}{l-1} \frac{1}{\delta(H)} \left(f_t(H)^2 + 2f_t(H)[f_t(H_k) - f_t(H)] + \right. \\
&\quad \left. 2f_t(H)[f_t(\hat{H}_k) - f_t(H)] + 2[f_t(H_k) - f_t(H)][f_t(\hat{H}_k) - f_t(H)] \right).
\end{aligned}$$

Der Faktor $p_c \frac{\delta(H)}{l-1}$ ist die Wahrscheinlichkeit, dass die Kreuzungsstelle zwischen die fixierten Positionen des Schemas fällt. Der Faktor $\frac{1}{\delta(H)}$ ist die

3.2 Modifikation von Hollands Schemensatz

Wahrscheinlichkeit, dass die Kreuzungsstelle k_0 genau an die gewünschte Stelle k fällt.

Der Wert von k kann zwischen h_1 und $h_{\delta(H)}$ variieren. Da die Kreuzungsstelle nur einmal ausgewählt wird, sind die Ereignisse $k = h_i$ und $k = h_j$ für $i \neq j$, ($i, j = h_1, \dots, h_{\delta(H)}$) disjunkt und die Wahrscheinlichkeiten der Ereignisse

$$P(\kappa(\{x_1, x_2\}, h_1) \cap H \neq \emptyset), \dots, P(\kappa(\{x_1, x_2\}, h_{\delta(H)}) \cap H \neq \emptyset)$$

können addiert werden. Also:

$$\begin{aligned} & P(\kappa(\{x_1, x_2\}, k) \cap H \neq \emptyset, k \in \{h_1, \dots, h_{\delta(H)}\}) = \\ & = p_c \frac{1}{l-1} \sum_{k=h_1}^{h_{\delta(H)}} \left(f_t(H)^2 + \right. \\ & \quad 2f_t(H)[f_t(H_k) - f_t(H)] + \\ & \quad 2f_t(H)[f_t(\hat{H}_k) - f_t(H)] + \\ & \quad \left. 2[f_t(H_k) - f_t(H)][f_t(\hat{H}_k) - f_t(H)] \right). \end{aligned}$$

Nun wird die zu erwartende Anzahl von Vertretern von H bei diesem Ereignis errechnet. Nur wenn zwei Vertreter von H miteinander gekreuzt werden, liegen beide resultierende Elemente in H . Bei den anderen drei Konstellationen kann nur eines der entstehenden Elemente in H liegen:

$$\begin{aligned} & E \left| \kappa(\{x_1, x_2\}, k) \cap H \neq \emptyset, k \in \{h_1, \dots, h_{\delta(H)}\} \right| = \\ & \quad \frac{p_c}{l-1} \sum_{k=h_1}^{h_{\delta(H)}} \left(2 \cdot f_t(H)^2 + 2f_t(H)[f_t(H_k) - f_t(H)] + \right. \\ & \quad \quad 2f_t(H)[f_t(\hat{H}_k) - f_t(H)] + \\ & \quad \quad \left. 2[f_t(H_k) - f_t(H)][f_t(\hat{H}_k) - f_t(H)] \right) \\ & = 2 \frac{p_c}{l-1} \sum_{k=h_1}^{h_{\delta(H)}} f_t(H_k) f_t(\hat{H}_k). \end{aligned} \tag{3.9}$$

Die Ereignisse $k \in \{h_1, \dots, h_{\delta(H)}\}$ und $k \notin \{h_1, \dots, h_{\delta(H)}\}$ sind ebenfalls disjunkt. Daher können die Wahrscheinlichkeiten und damit die Erwartungswerte für die zu erwartende Vertreteranzahl von H bei diesen Ereignissen ((3.8) und (3.9)) addiert werden. Damit erhält man einen Ausdruck für die zu erwartende Anzahl von Vertretern von H nach dem Anwenden des Kreuztauschoperators. Auf Grund von verbesserter Lesbarkeit variiert

3 Hollands Schemensatz und die Evolutionsgleichungen

der Wert der Kreuzungsstelle k nun zwischen 1 und $\delta(H)$. Innerhalb der Genome ist die Kreuzungsstelle dann $(h_1 - 1) + k$. Weiterhin werden insgesamt n Paare gekreuzt. Damit ist der Erwartungswert für die Anzahl der Vertreter von H nach dem Anwenden des Kreuztauschoperators:

$$\begin{aligned}
 E \left| O_K(S_t) \cap H \right| &= 2n f_t(H) \left(1 - p_c \frac{\delta(H)}{l-1} \right) + \\
 &\quad 2n \frac{p_c}{l-1} \sum_{k=1}^{\delta(H)} f_t(H_k) f_t(\hat{H}_k) \\
 &= 2n \left(f_t(H) \left(1 - p_c \frac{\delta(H)}{l-1} \right) + \frac{p_c}{l-1} \sum_{k=1}^{\delta(H)} f_t(H_k) f_t(\hat{H}_k) \right).
 \end{aligned} \tag{3.10}$$

(2) Betrachtet wird nun Szenario (2). Nach der Mutation eines Vertreters von H soll das resultierende Element ebenfalls in H liegen. Es wird angenommen, es existiert ein Vertreter x von H nach dem Kreuztausch. Die Wahrscheinlichkeit, dass aus der Mutation von x ein Vertreter von H entsteht, wurde bereits hergeleitet. Nach (3.4) ist

$$P(m(x) \in H) = (1 - p_m)^{o(H)}. \tag{3.11}$$

(3) Abschließend ist der letzte Fall zu betrachten. Aus der Mutation eines Elementes, welches nicht in H liegt, soll ein Vertreter von H resultieren. Um dieses Ereignis zu untersuchen, wird eine Klasse von Schemen \bar{H}_i eingeführt.

Definition 3.3 Die Menge \bar{H}_i ist eine Menge von Schemen, die sich an genau i Stellen von den fixierten Positionen von H unterscheiden. Ist also $H = (s_1, \dots, s_l)$, dann gilt:

$H' = (s'_1, \dots, s'_l) \in \bar{H}_i \Leftrightarrow$ für $s_k \neq *$ gilt $s'_k = 1 - s_k$ für $k = k_1, \dots, k_i$ und $s'_m = s_m$ für $m = k'_1, \dots, k'_{o(H)-i}$.

Ein Schema $H' \in \bar{H}_i$ kann also aus H konstruiert werden, indem i beliebige Allele an den fixierten Positionen von H negiert werden. In Abbildung 3.2 sind einige Beispiele für Schemen, die in \bar{H}_i liegen, graphisch dargestellt. Das Schema H' besitzt die gleiche Ordnung und Länge wie das Schema H :

$$\begin{aligned}
 o(H') &= o(H) \\
 \delta(H') &= \delta(H).
 \end{aligned}$$

3.2 Modifikation von Hollands Schemensatz

Schema H	0	*	*	1	0	1	1	*	*
\bar{H}_2	0	*	*	1	1	0	1	*	*
	1	*	*	1	1	1	1	*	*
\bar{H}_4	0	*	*	0	1	0	0	*	*

Abbildung 3.2: Beispiele für Schemen aus den Klassen \bar{H}_i für $i = 2$ und 4 .

Es wird angenommen, dass nach dem Anwenden des Kreuztauschoperators ein Element $x_{H_i} \in H' \in \bar{H}_i$ existiert, welches sich an genau i Stellen von den Allelen der fixierten Positionen des Schemas H unterscheidet.

Das Element x_{H_i} mutiert zu einem Vertreter von H , wenn an genau den i Positionen, an denen H' sich von H unterscheidet, Mutationen stattfinden. Die restlichen Positionen von x_{H_i} , deren Allele mit denen von H übereinstimmen, dürfen dann nicht mutiert werden. Die Wahrscheinlichkeit, dass i Mutationen stattfinden und $(o(H) - i)$ Mutationen nicht, ist:

$$P(i \text{ Mutationen an best. fixierten Pos. von } H) = p_m^i (1 - p_m)^{o(H) - i}. \quad (3.12)$$

Es wurde angenommen, dass ein solches Element x_{H_i} in der Menge der Individuen nach Anwenden des Kreuztauschoperators vorhanden ist. Die Wahrscheinlichkeit $P(x_{H_i} \in O_K(S_t))$ ist jedoch nicht nur abhängig von der Fitness $f_t(x_{H_i})$, der Länge und Ordnung von H' , sondern auch von der derzeit vorhandenen Population. Für die Wahrscheinlichkeit, dass nach dem Anwenden des Kreuztauschoperators Vertreter von H in $O_K(S_t)$ sind, die sich an genau i Stellen von H unterscheiden, müssen daher alle $\binom{o(H)}{i}$ Schemen $H' \in \bar{H}_i$ betrachtet werden:

$$P(x_{H_i} \in O_K(S_t)) = \sum_{H' \in \bar{H}_i} P(O_K(S_t) \cap H' \neq \emptyset).$$

Nun kann die Wahrscheinlichkeit angegeben werden, dass ein Vertreter eines Schemas $H' \in \bar{H}_i$ vorhanden ist und zu einem Vertreter von H mutiert:

$$\begin{aligned} P(m(x_{H_i}) \in H) &= P(i \text{ Mut. best. Pos. von } H) P(x_{H_i} \in O_K(S_t)) \\ &= p_m^i (1 - p_m)^{o(H) - i} \sum_{H' \in \bar{H}_i} P(O_K(S_t) \cap H' \neq \emptyset). \end{aligned} \quad (3.13)$$

Der Wert von i kann zwischen 1 und $o(H)$ variieren. So erhält man die Wahrscheinlichkeit, dass ein Vertreter von H nach der Mutation eines Elementes, welches nicht in H liegt, resultiert:

3 Hollands Schemensatz und die Evolutionsgleichungen

$$\begin{aligned}
 & P(m(x) \in H \text{ mit } x \notin H) = \\
 & \sum_{i=1}^{o(H)} p_m^i (1 - p_m)^{o(H)-i} \sum_{H' \in \hat{H}_i} P(O_K(S_t) \cap H' \neq \emptyset). \quad (3.14)
 \end{aligned}$$

Da aus der Mutation von einem Element nur ein Element resultiert, kann maximal ein Vertreter von H nach der Mutation vorhanden sein. Der Mutationsoperator O_M wird auf $2n$ Elemente angewendet, daher ist die erwartete Anzahl von Vertretern von H , die aus der Mutation eines Elementes, das nicht in H liegt, resultiert:

$$\begin{aligned}
 E \left| \{x : x \notin H \text{ und } m(x) \in H\} \right| = \\
 2n \sum_{i=1}^{o(H)} p_m^i (1 - p_m)^{o(H)-i} \sum_{H' \in \hat{H}_i} P(O_K(S_t) \cap H' \neq \emptyset). \quad (3.15)
 \end{aligned}$$

Nun können (3.10), (3.11) und (3.15) zusammengesetzt werden. Man erhält folgenden Satz:

Satz 3.4 (Modifizierter Schemensatz) *In einem einfachen genetischen Algorithmus ist die erwartete Anzahl von Vertretern eines Schemas H in der Folgegeneration:*

$$\begin{aligned}
 E \left| P_{t+1} \cap H \right| & =: E_S \\
 & = 2n f_t(H) \left(1 - p_c \frac{\delta(H)}{l-1} \right) (1 - p_m)^{o(H)} + \\
 & \quad 2n \frac{p_c}{l-1} \sum_{k=1}^{\delta(H)} f_t(H_k) f_t(\hat{H}_k) (1 - p_m)^{o(H)} + \\
 & \quad 2n \sum_{i=1}^{o(H)} p_m^i (1 - p_m)^{o(H)-i} \sum_{H' \in \hat{H}_i} P(O_K(S_t) \cap H' \neq \emptyset). \quad (3.16)
 \end{aligned}$$

Der modifizierte Schemensatz bietet einen exakten Ausdruck für den Erwartungswert der Anzahl der Vertreter eines Schemas in der Folgepopulation. Er wird auch Evolutionsgleichung genannt. Somit sind genauere Untersuchungen an der Entwicklung der Population eines genetischen Algorithmus möglich als bisher. Es könnten nun zum Beispiel Abschätzungen für das erstmalige Auftreten des Optimums in der Population bei gegebenen Parameterbelegungen berechnet werden.

Weiterhin kann die Evolutionsgleichung für die Optimierung der Parameter eines einfachen genetischen Algorithmus genutzt werden. Bisher war dies mit dem Schemensatz nur eingeschränkt möglich.

3.3 Stephens Evolutionsgleichung

Natürlich haben sich auch andere Forscher mit dem Schemensatz und seiner Verbesserung auseinandergesetzt. Und so ist es nicht verwunderlich, dass es weitere Evolutionsgleichungen gibt. C. R. Stephens hat seine erstmals 1996 veröffentlicht [SH98], [SW99]. Stephens stellte nicht nur die Evolutionsgleichungen für Schemen auf, sondern auch für Zeichenketten und beliebige genetische Operatoren [SH98], [Ste01]. Für flache Fitnesslandschaften ohne Mutation und mit 1-Punkt Kreuztausch fand er explizite Lösungen, die nicht nur das asymptotische Verhalten liefern, sondern auch alle Zwischenzustände [Ste01].

Die Herleitung der Evolutionsgleichung (3.16) in dieser Diplomarbeit fand jedoch in Unkenntnis von Stephens Veröffentlichungen statt. Die Gleichungen lassen sich leicht ineinander überführen. Die Evolutionsgleichung von Stephens ([SH98], [SW99], [Ste01]) lautet:

$$\begin{aligned}
 P(H, t+1) &= P_m(H \rightarrow H)P_c(H, t) + \sum_{H_i \neq H} P_m(H_i \rightarrow H)P_c(H_i, t) \\
 P_c(H, t) &= P'(H, t) - \frac{p_c}{l-1} \sum_{k=1}^{\delta(H)} (P'(H, t) - P'(H_L, t)P'(H_R, t)).
 \end{aligned}
 \tag{3.17}$$

Dabei ist $P(H, t+1)$ die Wahrscheinlichkeit, mit der Schema H in P_{t+1} auftritt und $P_m(H_a \rightarrow H_b)$ die Wahrscheinlichkeit, dass Schema H_a in Schema H_b mutiert wird. Die Wahrscheinlichkeit, ein Schema H zur Rekombination auszuwählen, ist mit $P'(H, t)$ bezeichnet. Der Term $P_c(H, t)$ bezeichnet die Wahrscheinlichkeit, mit der H aus der Rekombination die auf P_t durchgeführt wird, hervorgeht. Die Schemen H_L und H_R sind diejenigen Teilschemen von H , die bis zur Kreuzungsstelle k mit H übereinstimmen (H_L), bzw. ab der Kreuzungsstelle mit H übereinstimmen (H_R), das heißt:

$$\begin{aligned}
 H_L &= H_k \\
 H_R &= \hat{H}_k.
 \end{aligned}$$

Für die bitweise Mutation und fitnessproportionale Selektion lauten folgende Terme konkret:

$$\begin{aligned}
 P_m(H \rightarrow H) &= (1 - p_m)^l \\
 P_m(H_i \rightarrow H) &= p_m^{d_{H, H_i}^H} (1 - p_m)^{o(H) - d_{H, H_i}^H} \\
 P'(H, t) &= f_t(H),
 \end{aligned}$$

wobei d_{H, H_i}^H die Hamming-Distanz der Schemen H und H_i ist. Die Hamming-Distanz zwischen eines fixierten und unfixierten Bits wird dabei als 1 definiert. Damit kann man $P_c(H, t)$ umformen:

3 Hollands Schemensatz und die Evolutionsgleichungen

$$\begin{aligned}
P_c(H, t) &= f_t(H) - \frac{p_c}{l-1} \sum_{k=1}^{\delta(H)} (f_t(H) - f_t(H_k) f_t(\hat{H}_k)) \\
&= f_t(H) - p_c \frac{\delta(H)}{l-1} f_t(H) + \frac{p_c}{l-1} \sum_{k=1}^{\delta(H)} f_t(H_k) f_t(\hat{H}_k) \\
&= f_t(H) \left(1 - p_c \frac{\delta(H)}{l-1} \right) + \frac{p_c}{l-1} \sum_{k=1}^{\delta(H)} f_t(H_k) f_t(\hat{H}_k)
\end{aligned}$$

Der Ausdruck für die Konstruktion durch Mutation ist gleich:

$$\begin{aligned}
\sum_{H_i \neq H} P_m(H_i \rightarrow H) P_c(H_i, t) &= \sum_{H_i \neq H} P_m(H_i \rightarrow H) P(O_K(S_t) \cap H_i \neq \emptyset) \\
&= \sum_{i=1}^{o(H)} P_m(\bar{H}_i \rightarrow H) \sum_{H' \in \bar{H}_i} P(O_K(S_t) \cap H' \neq \emptyset) \\
&= \sum_{i=1}^{o(H)} p_m^{d_{\bar{H}_i, H}^H} (1 - p_m)^{o(H) - d_{\bar{H}_i, H}^H} \\
&\quad \cdot \sum_{H' \in \bar{H}_i} P(O_K(S_t) \cap H' \neq \emptyset) \\
&= \sum_{i=1}^{o(H)} p_m^i (1 - p_m)^{o(H) - i} \sum_{H' \in \bar{H}_i} P(O_K(S_t) \cap H' \neq \emptyset).
\end{aligned}$$

Es gilt also:

$$\frac{1}{2n} E_S = P(H, t + 1).$$

Damit stimmen die Evolutionsgleichungen inhaltlich überein.

3.4 Verallgemeinerung der Evolutionsgleichung auf beliebige genetische Operatoren

Nur ein kleiner Schritt ist nötig, um die Evolutionsgleichung auf beliebige genetische Operatoren zu verallgemeinern. In der Literatur ist eine solche Verallgemeinerung zum Beispiel in [SH98] und [Ste01] zu finden. Es ist günstig von Stephens Evolutionsgleichung (3.17) auszugehen.

Die Ausdrücke für den Mutationsoperator $P_m(H_a \rightarrow H_a)$ und den Selektionsoperator $P'(H)$ sind bereits für beliebige Operatoren verallgemeinert. Es bleibt die Aufgabe, einen allgemeinen Ausdruck für den Rekombinationsoperator zu finden. Das Ziel ist ein Rekombinationsoperator, der

3.4 Verallgemeinerung der Evolutionsgleichung auf beliebige genetische Operatoren



Abbildung 3.3: Rekombination mit der Maske 10010.

verschiedenen Masken zulässt. Eine Rekombinationsmaske ist eine binäre Zeichenkette, dessen i -tes Bit die Herkunft des i -ten Bits der Nachkommen bestimmt. Eine 1 bedeutet den Austausch der Allele an der betroffenen Position. Abbildung 3.3 zeigt die Genome vor und nach der Kreuzung mit der Rekombinationsmaske 10010. Der erste Nachkomme erhält das erste und vierte Bit vom zweiten Elter, die restlichen Bits vom ersten Elter. Für den zweiten Nachkommen verhält es sich genau umgedreht.

Die Menge der Rekombinationsmasken wird mit M_c bezeichnet, ein Element von M_c mit M . Die Wahrscheinlichkeit, dass die Rekombination mit der Maske M ausgeführt wird, ist mit $p(M)$ gekennzeichnet. Damit lässt sich die Evolutionsgleichung (3.17) allgemein wie folgt darstellen:

$$\begin{aligned}
 P(H, t+1) &= P_m(H \rightarrow H)P_c(H, t) + \sum_{H_i \neq H} P_m(H_i \rightarrow H)P_c(H_i, t) \\
 &= \sum_{H_i} P_m(H_i \rightarrow H)P_c(H_i, t) \\
 P_c(H, t) &= P'(H, t) - p_c \sum_M p(M) (P'(H, t) - P'(H_L, t)P'(H_R, t)).
 \end{aligned} \tag{3.18}$$

Dabei symbolisiert H_L bzw. H_R dasjenige Schema, das mit H an den Stellen übereinstimmt, an denen die Maskenbits 1 bzw 0 sind.

Die Evolutionsgleichung kann auch für Zeichenketten angegeben werden. Sie lautet für eine Zeichenkette I [SH98]:

$$\begin{aligned}
 P(I, t+1) &= P_m(I \rightarrow I)P_c(I, t) + \sum_{J \neq I} P_m(J \rightarrow I)P_c(J, t) \\
 &= \sum_J P_m(J \rightarrow I)P_c(J, t) \\
 P_c(I, t) &= P'(I, t) \\
 &\quad - p_c \sum_M p(M) \sum_{J, K \in P_t} (P'(I, t) - \lambda_I^{JK}(M)P'(J, t)P'(K, t)),
 \end{aligned} \tag{3.19}$$

dabei ist $\lambda_I^{JK}(M)$ genau dann 1, wenn der erste Nachkomme der aus der Rekombination zwischen den Individuen J und K mit der Maske M hervorgehenden Nachkommen I ist, und ansonsten 0.

3.5 Homogenisierung der Population

Mit voranschreitender Generationzahl im Genetischen Algorithmus wird die Population homogener, das heißt, die Genotypen der Populationselemente gleichen sich an. Die Populationselemente liegen also weniger verstreut im Suchraum, sondern haben durchschnittlich eine geringere Hamming-Distanz zueinander als die Elemente einer zufälligen Population. Man spricht bei einer homogenen Population auch von einer geordneten Population.

Untersucht man die Evolutionsgleichung (3.18), so erkennt man, dass die Homogenisierung der Population durch die Selektion und die Rekombination hervorgerufen wird. Die Mutation hat die Aufgabe, immer wieder neue Genotypen in die Population einzuführen, damit die Population nicht auf einem lokalen Maximum hängenbleibt. Die Mutation wirkt also der Homogenisierung entgegen.

Die Homogenisierung des Genpools, also der Gesamtheit der Genotypen in der Population, wird erreicht, wenn isolierte Elemente aussortiert werden. Ein Populationselement ist isoliert, wenn seine Häufigkeit in der Population sehr gering ist oder wenn es wenig oder keine Populationselemente in seiner Umgebung gibt. Das heißt die Hamming-Distanz zu anderen Populationselementen ist groß. Damit gibt es auch weniger Elemente in der Population, die mit dem isolierten Element Teilschemen gemeinsam haben. Ein Schema heißt isoliert, wenn seine Vertreter in der Population isoliert sind.

Betrachtet man zwei Elemente oder auch Schemen H und H_{iso} , wovon das letztere isoliert sein soll, die dieselben Fitnesswerte haben und mit derselben Häufigkeit in der Population vertreten sind:

$$\begin{aligned} f_t(H) &= f_t(H_{iso}) \\ P(H, t) &= P(H_{iso}, t), \end{aligned}$$

so folgt zum Beispiel mit der fitnessproportionalen Selektion

$$P'(H, t) = P'(H_{iso}, t).$$

Die Selektion verändert die relativen Häufigkeiten der beiden Schemen in der Folgepopulation nicht.

Untersucht man die Wahrscheinlichkeit, mit der H und H_{iso} nach der Rekombination vorhanden sind, so sieht man, dass

$$P_c(H, t) > P_c(H_{iso}, t),$$

denn

$$P'(H_L, t)P'(H_R, t) > P'(H_{iso_L}, t)P'(H_{iso_R}, t).$$

Das Schema H_{iso} ist nach Voraussetzung isoliert, demnach sind $P'(H_{iso_L}, t)$ und $P'(H_{iso_R}, t)$ kleiner als $P'(H_L, t)$ und $P'(H_R, t)$.

Die Rekombination verändert also die relativen Häufigkeiten von H und H_{iso} zu Gunsten des weniger isolierten Elements, trotz identischer Fitness- und Häufigkeitwerte.

Im nächsten Schritt der Selektion gilt dann

$$P'(H, t + 1) > P'(H_{iso}, t + 1),$$

da

$$f_{t+1}(H)P(H, t + 1)2n > f_{t+1}(H_{iso})P(H_{iso}, t + 1)2n.$$

Die Selektion verringert den Anteil des isolierten Elementes in der Population weiter und die Population wird homogener, obwohl die Selektion nur auf dem Phänotyp agiert und nicht auf dem Genotyp.

Das Wechselspiel zwischen Selektion und Rekombination sorgt also für die Homogenisierung, das heißt die Angleichung des Erbguts in der Population. Die Mutation wirkt dem entgegen.

3.6 Zusammenfassung

Zum Abschluss dieses Kapitels sollen nochmals die wesentlichen Aspekte zusammengefasst werden. Das Kapitel begann mit der Herleitung des Schemensatzes von Holland [Hol75] und der Verbesserung seiner Abschätzung in eine Gleichheitsaussage.

Es wurde gezeigt, dass der modifizierte Schemensatz (3.16) mit der Evolutionsgleichung von Stephens [SH98] äquivalent ist. Abschließend wurde dieser auf beliebige genetische Operatoren verallgemeinert [SH98] und gezeigt, dass die Homogenisierung des Genpools durch das Wechselspiel von Selektion und Rekombination bedingt ist.

3 Hollands Schemensatz und die Evolutionsgleichungen

4 Die Symmetrie als Fitnessfunktion und die gerichtete Rekombination

Ziel dieses Kapitels ist die Einführung einer neuen Fitnessfunktion und eines neuen Rekombinationsoperators. Die Evolution in einem Genetischen Algorithmus kann damit realistischer aber dennoch mathematisch behandelbar abgebildet werden.

Weiterhin wird im Laufe des Kapitels die Fitnesslandschaft, die zur Symmetrie gehört, analysiert. Es wird untersucht, wie die genetischen Operatoren auf dieser Landschaft agieren. Die Wirkung des neuen Rekombinationsoperators wird ebenfalls auf der Fitnesslandschaft analysiert. Die Evolutionsgleichung wird an diese beiden Neuerungen, Fitnessfunktion und Rekombination, angepasst. Zunächst wird die Fitnessfunktion vorgestellt. Der neue Rekombinationsoperator wird in Abschnitt 4.5 behandelt.

In der Natur lassen die meisten Organismen auffällige Symmetrien erkennen. Auch der Aufbau von Mineralien und Kristallen ist hauptsächlich symmetrisch. Diese Vorliebe der Natur für Symmetrie soll in einen Genetischen Algorithmus integriert werden. Daher wird die neue Fitnessfunktion ein Maß der Spiegelsymmetrie der Elemente sein. Je mehr symmetrische Bits in der Zeichenkette eines Elementes vorhanden sind, desto höher ist seine Fitness. Es wird das erste Bit mit dem letzten Bit verglichen, das zweite Bit mit dem vorletzten und so weiter. Dabei heißt ein Bit symmetrisch, wenn sein Allel mit dem entsprechendem Allel in der anderen Zeichenkettenhälfte übereinstimmt. Ein Bit heißt unsymmetrisch im anderen Fall. In der Zeichenkette 011010 wäre also das erste und das zweite Bit symmetrisch, das dritte Bit hingegen nicht. Bei einer Zeichenkettenlänge von l gibt es also maximal $\frac{l}{2}$ symmetrische oder $\frac{l}{2}$ unsymmetrische Bits. Die Summe aller symmetrischen und unsymmetrischen Bits beträgt dann $\frac{l}{2}$.

Die Spiegelsymmetrie als Bewertungsfunktion nähert die intrinsische Fitness an. Für die intrinsische Fitness existieren mehrere Definitionen, unter anderem wird sie in manchen Veröffentlichungen äquivalent zur effektiven Fitness gebraucht, welche in Kapitel 5 vorgestellt wird. In dieser Arbeit bedeutet intrinsische Fitness, dass der Fitnesswert nur von einer, dem Individuum eigenen Eigenschaft abhängt, nicht jedoch von äußeren Faktoren, wie zum Beispiel seiner Umwelt, der Zusammensetzung der Population, seinem Verhalten oder der Größe seines Funktionswertes in einer Zielfunktion. Auch bei der natürlichen Selektion wird der Fitnesswert, sprich die Fortpflanzungschance, nicht ausschließlich durch eine äußere Funktion bestimmt, sondern durch ein kompliziertes Zusammenspiel von mehreren äußeren Faktoren und individuellen Eigenschaften.

4 Die Symmetrie als Fitnessfunktion und die gerichtete Rekombination

Spiegelsymmetrie kann auf mehrere Arten realisiert werden. So sind zum Beispiel die beiden Elemente 101101 und 000000 im gleichen Maß symmetrisch, haben aber einen sehr unterschiedlichen Genotyp. Diese Multimodalität zählt ebenfalls zu den Vorteilen, da auch in der Natur die Anpassung an die Umwelt auf höchst unterschiedliche Weise realisiert wird und die verschiedenen Organismen gleichberechtigt nebeneinander existieren. Es ist zu hoffen, dass mit der Symmetrie die Entstehung von mehreren Arten in den Simulationen nachgestellt werden kann. Ergebnisse dazu sind in Kapitel 7 zu finden.

Eine weitere natürliche Begebenheit kann durch die Symmetrie modelliert werden: die sogenannte neutrale Evolution. Allerdings nur unter der Voraussetzung, dass die Zeichenkettenlänge ungerade ist. Es gibt dann in der Fitnesslandschaft, die zur Spiegelsymmetrie gehört, mehrere neutrale Becken. Ein neutrales Becken ist eine Menge von nebeneinander liegenden Elementen, das heißt sie unterscheiden sich jeweils nur um ein Bit, die dergleichen Fitnesswert haben. Dies bedeutet, dass trotz unterschiedlicher Genotypen kein Selektionsvorteil eines Elementes gegenüber eines anderen innerhalb eines neutralen Beckens besteht. Diese Neutralität wird durch das mittlere Bit in den Zeichenketten verursacht, denn dessen Ausprägung beeinflusst in keiner Weise den Fitnesswert. In Kapitel 5 wird auf die neutrale Evolution eingegangen. Dort wird man sehen, dass neutrale Becken nicht auf beliebigen Wege verlassen werden, sondern in der Richtung, in der der effektive Fitnessgradient am größten ist.

Die Symmetrie als Bewertungsfunktion modelliert nicht nur die natürliche Selektion realistischer als eine zu maximierende Zielfunktion, sondern ist auch noch hinreichend einfach, um mathematische Untersuchungen zu ermöglichen.

4.1 Die Fitnesslandschaft

Im Folgenden soll die zugehörige Fitnesslandschaft analysiert werden. Die Fitnesslandschaft, also der Graph der Fitnessfunktion über dem Suchraum, hat $2^{\frac{l}{2}}$ Maxima, bei einer Stringlänge von l und geradem l . Die Multimodalität steigt also mit $\frac{l}{2}$. Wenn l ungerade ist, so gibt es $2^{\frac{l-1}{2}} \cdot 2 = 2^{\frac{l+1}{2}}$ Maxima, aber dieses Detail soll im Folgenden vernachlässigt werden, da es keine zusätzlichen Erkenntnisse bringt und die Untersuchungen erschwert.

Besteht ein Element nur aus symmetrischen Bits, so besitzt es den höchsten Fitnesswert von $f_{max} = \frac{l}{2}$. Die Anzahl der Elemente im gesamten Suchraum mit einer Fitness von k , $k \in \{0, \dots, f_{max}\}$ ist

$$\binom{\frac{l}{2}}{k} 2^{\frac{l}{2}-k} 2^k = \binom{\frac{l}{2}}{k} 2^{\frac{l}{2}}.$$

Die Wahrscheinlichkeit, dass ein Bit symmetrisch ist, ist $\frac{1}{2}$. Die durch-

4.2 Wirkung der Mutation in der Fitnesslandschaft

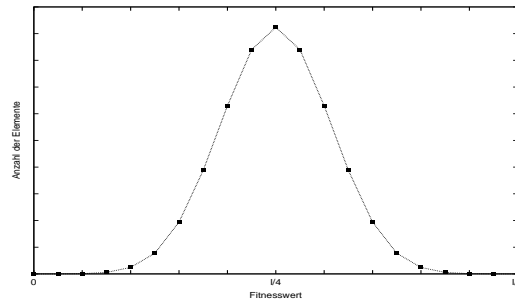


Abbildung 4.1: Verteilung der Fitnesswerte über dem Suchraum

schnittliche Symmetrie einer beliebigen Zeichenkette ist damit

$$\frac{\frac{l}{2} \frac{1}{2}}{l} = \frac{1}{4}. \quad (4.1)$$

Die Fitnesswerte sind über dem Suchraum binomial verteilt. Die Verteilung besitzt die Parameter $p = \frac{1}{2}$ und $n = \frac{l}{2}$ (siehe Abbildung 4.1).

Der Hamming-Abstand eines Elementes zu einem vollsymmetrischen Element liefert Rückschlüsse auf seine Fitness. Diese steigt mit geringer werdenden Hamming-Abstand. Jedes vollsymmetrische Element ist von l Elementen umgeben, die einen um 1 niedrigeren Symmetriewert besitzen. Hat ein Element eine Symmetrie bzw. Fitness von k , $k \in \{0, \dots, \frac{l}{2}\}$, so ist es umgeben von $2(\frac{l}{2} - k)$ fitteren und $2k$ weniger fitten Individuen. Dabei kann sich der Fitnesswert nur um genau 1 unterscheiden. An jedem Punkt des Suchraums steigt oder sinkt also der Fitnessgradient. Es gibt keine lokalen Extrema. Globale Maxima werden von den vollsymmetrischen Elementen gebildet, globale Minima von den komplett unsymmetrischen. Es gibt genau soviele globale Maxima wie Minima.

4.2 Wirkung der Mutation in der Fitnesslandschaft

Es ist notwendig, die Auswirkung der Mutation auf den Symmetriewert eines Elementes zu untersuchen. Hierfür werden im Folgenden Formeln hergeleitet, die die Wahrscheinlichkeit der Veränderung der Symmetrie eines Elementes durch die Mutation angeben.

Wird ein Bit in einer Zeichenkette geändert, so steigt oder sinkt die Fitness genau um 1. Ist der Fitnesswert eines Elementes gleich s , so kann die Wahrscheinlichkeitsverteilung für den Fitnesswert nach k Mutationen angegeben werden:

4 Die Symmetrie als Fitnessfunktion und die gerichtete Rekombination

$$\begin{aligned}
 &P(\text{Erhöhung des Fitnesswertes um } m) = \\
 &\frac{1}{\binom{l}{k}} \sum_{n_1+n_2+n_3=\frac{k-m}{2}} 2^{2n_2+m} \binom{u}{n_1+n_2+m} \binom{n_1+n_2+m}{n_1} \binom{s}{n_2+n_3} \binom{n_2+n_3}{n_3} \\
 &P(\text{Verringerung des Fitnesswertes um } m) = \\
 &\frac{1}{\binom{l}{k}} \sum_{n_1+n_2+n_3=\frac{k-m}{2}} 2^{2n_2+m} \binom{u}{n_1+n_2} \binom{n_1+n_2}{n_1} \binom{s}{n_2+n_3+m} \binom{n_2+n_3+m}{n_3}.
 \end{aligned} \tag{4.2}$$

Dabei wird mit u die Anzahl der unsymmetrischen Bits und mit s die Anzahl der symmetrischen Bits bezeichnet. Es gilt also $u+s = \frac{l}{2}$. Die Summe läuft über alle positiven ganzen Zahlen n_1 , n_2 und n_3 , die addiert $\frac{k-m}{2}$ ergeben. Dabei ist n_1 die Anzahl der Mutationen, bei denen beide Allele eines unsymmetrischen Bits mutiert werden und n_3 die Anzahl der Mutationen, bei denen beide Allele eines symmetrischen Bits mutiert werden. Die Zahl n_2 steht für die Anzahl von jeweils zwei Mutationen, bei denen je ein unsymmetrisches und ein symmetrisches Bit mutiert werden. Für ein gerades k kann die Variable m nur Werte in $\{0, 2, \dots, k-2, k\}$ annehmen. Ist k ungerade, so gilt $m \in \{1, 3, \dots, k-2, k\}$.

Die Formeln (4.2) sind ein Spezialfall der hypergeometrischen Verteilung. Um den Fitnesswert um m zu erhöhen, müssen mindestens m Mutationen an den unsymmetrischen Bits stattfinden. Außerdem müssen dann die restlichen $k-m$ Mutationen ohne Auswirkung bleiben. Das heißt $k-m$ muss durch zwei teilbar sein, da sich nur jeweils zwei Mutationen aufheben können.

Wann heben sich zwei Mutationen bezüglich ihrer Fitnessänderung auf? Dafür gibt es drei verschiedene Konstellationen. Zum Einen können die beiden Mutationen auf beide Bitstellen eines unsymmetrischen Bits bzw. eines symmetrischen Bits fallen (n_1 bzw. n_3). So bliebe das unsymmetrische Bit unsymmetrisch und das symmetrische Bit bliebe symmetrisch. Zum Dritten kann eine Mutation an einem symmetrischen Bit durch eine Mutation an einem unsymmetrischen Bit in der Wertigkeit wieder aufgehoben werden (n_2). Die Änderung des symmetrischen Bits erwirkt eine Senkung des Fitnesswertes um 1 und die Änderung des unsymmetrischen Bits eine Erhöhung um 1. Abbildung 4.2 verdeutlicht die drei Fälle an einem Beispiel.

Die Variablen n_1 , n_2 und n_3 werden eingeführt, um wiederzugeben, auf welche Weise sich je zwei Mutationen aufheben: auf die erste, zweite oder dritte Weise. Zum Beispiel gibt n_1 an, dass sich n_1 Zweier-Mutationen aufheben, indem beide Stellen eines unsymmetrischen Bits getroffen wurden. Ebenso steht n_3 für die Aufhebung durch Mutationen beider Stellen eines symmetrischen Bits. Es gibt n_2 Zweier-Mutationen, bei denen je ein symmetrisches und ein unsymmetrisches Bit getroffen wurde. Alle möglichen Kombinationen der Belegung von n_1 , n_2 und n_3 werden in der Summe

4.2 Wirkung der Mutation in der Fitnesslandschaft

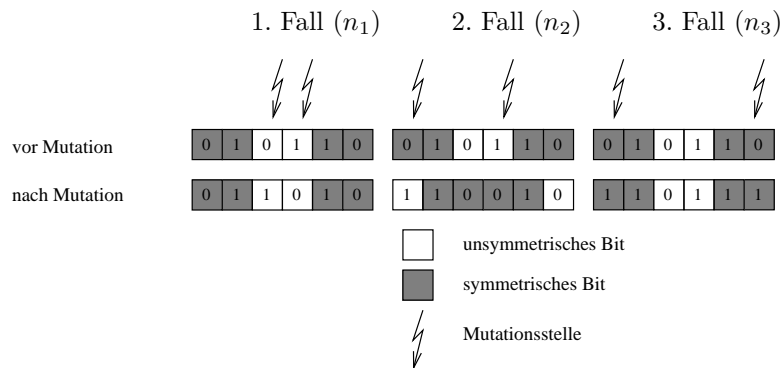


Abbildung 4.2: Aufhebung der Fitnessveränderung durch je zwei Mutationen

durchlaufen. Da die n_i ($i = 1, 2, 3$) die Anzahl von Zweier-Mutationen angeben, muss ihre Summe genau $\frac{k-m}{2}$ ergeben.

Betrachtet werden nun die möglichen Kombinationen. Der Term in (4.2)

$$2^{n_2+m} \binom{u}{n_1 + n_2 + m} \binom{n_1 + n_2 + m}{n_1}$$

gibt die Anzahl der Möglichkeiten an, mit denen man auf der einen Zeichenkettenhälfte $n_1 + n_2 + m$ Mutationsstellen aus den unsymmetrischen Bits auswählen kann und n_1 Stellen auf der anderen Zeichenkettenhälfte aus denjenigen unsymmetrischen Bits, deren Gegenstück bereits mutiert wurde. Bei den m Mutationen, die zur Fitnesserhöhung beitragen sowie den n_2 -Mutationen ist es egal, auf welche Zeichenkettenhälfte die Mutationen fallen, daher wird der Faktor 2^{n_2+m} dazu multipliziert.

Der Term

$$2^{n_2} \binom{s}{n_2 + n_3} \binom{n_2 + n_3}{n_3}$$

gibt die Anzahl der Möglichkeiten an, mit denen $n_2 + n_3$ Mutationsstellen aus den symmetrischen Bits der einen Zeichenkettenhälfte ausgewählt werden können. Die Anzahl der Stellen aus den symmetrischen Bits der anderen Hälfte, deren Gegenstücke ebenfalls mutiert werden ist n_3 . Der Faktor 2^{n_2} kommt noch hinzu, da es auch hier wieder keine Rolle spielt, auf welcher Zeichenkettenhälfte die n_2 -Mutationen stattfinden.

Die Formel zur Verringerung der Fitness erklärt sich auf die gleiche Weise, nur dass nun m Mutationen an den symmetrischen Bits die Fitnessveränderung hervorrufen.

Das folgende Beispiel gibt die Wahrscheinlichkeitsverteilung für $l = 12$ und $k = 5$ an. Dabei wird ein Element untersucht, das vier symmetrische Bits und zwei unsymmetrische Bits besitzt: $s = 4$, $u = 2$.

4 Die Symmetrie als Fitnessfunktion und die gerichtete Rekombination

$$\begin{aligned} \binom{12}{5} P(\text{Erhöhung um 5}) &= 2^5 \binom{2}{5} \quad (n_1 = n_2 = n_3 = 0) \\ &= 0 \\ \binom{12}{5} P(\text{Verringerung um 5}) &= 2^5 \binom{4}{5} \quad (n_1 = n_2 = n_3 = 0) \\ &= 0 \end{aligned}$$

$$\begin{aligned} \binom{12}{5} P(\text{Erhöhung um 3}) &= 2^3 \binom{2}{4} \binom{4}{1} \binom{4}{0} \binom{0}{0} \quad (n_1 = 1) \\ &\quad + 2^5 \binom{2}{4} \binom{4}{0} \binom{4}{1} \binom{1}{0} \quad (n_2 = 1) \\ &\quad + 2^3 \binom{2}{3} \binom{3}{0} \binom{4}{1} \binom{1}{1} \quad (n_3 = 1) \\ &= 0 \end{aligned}$$

$$\begin{aligned} \binom{12}{5} P(\text{Verringerung um 3}) &= 2^3 \binom{2}{1} \binom{1}{1} \binom{4}{3} \binom{3}{0} \quad (n_1 = 1) \\ &\quad + 2^5 \binom{2}{1} \binom{1}{0} \binom{4}{4} \binom{4}{0} \quad (n_2 = 1) \\ &\quad + 2^3 \binom{2}{0} \binom{0}{0} \binom{4}{4} \binom{4}{1} \quad (n_3 = 1) \\ &= 160 \end{aligned}$$

$$\begin{aligned} \binom{12}{5} P(\text{Erhöhung um 1}) &= 2 \binom{2}{3} \binom{3}{2} \binom{4}{0} \binom{0}{0} \quad (n_1 = 2) \\ &\quad + 2^5 \binom{2}{3} \binom{3}{0} \binom{4}{2} \binom{2}{0} \quad (n_2 = 2) \\ &\quad + 2 \binom{2}{1} \binom{1}{0} \binom{4}{2} \binom{2}{2} \quad (n_3 = 2) \\ &\quad + 2^3 \binom{2}{3} \binom{3}{1} \binom{4}{1} \binom{1}{0} \quad (n_1, n_2 = 1) \\ &\quad + 2 \binom{2}{2} \binom{2}{1} \binom{4}{1} \binom{1}{1} \quad (n_1, n_3 = 1) \\ &\quad + 2^3 \binom{2}{2} \binom{2}{0} \binom{4}{2} \binom{2}{1} \quad (n_2, n_3 = 1) \\ &= 136 \end{aligned}$$

$$\binom{12}{5} P(\text{Verringerung um 1}) = 496.$$

4.3 Wirkung der Rekombination in der Fitnesslandschaft

Nachdem nun klar ist, wie die Mutation auf der neuen Fitnesslandschaft agiert, bleibt es zu untersuchen, wie es sich mit der Rekombination verhält. Diese wird im nächsten Abschnitt behandelt.

4.3 Wirkung der Rekombination in der Fitnesslandschaft

Das Resultat einer Kreuzung ist nicht trivial und kann nicht so einfach wiedergegeben werden, wie es bei der Mutation der Fall ist. Die Wirkung der Rekombination hängt nicht nur von den Allelen der Eltern ab, unterscheiden sie sich oder nicht, sondern auch davon, welche Maske verwendet wird und ob ein betreffendes Bit symmetrisch oder unsymmetrisch ist.

Die Rekombination ändert die Hamming-Distanz zweier Eltern nicht. Bei einer Kreuzung werden Allele nur ausgetauscht, nicht aber in ihrer Häufigkeit in den Bits verändert. Daher haben die beiden Nachkommen genau dieselbe Hamming-Distanz zueinander wie ihre Eltern. Man kann aber davon ausgehen, dass mit größer werdenden Hamming-Abstand der Eltern das Intervall, in dem die Fitnesswerte der Kinder liegen können, immer größer wird. Ein großer Hamming-Abstand bedeutet mehr Möglichkeiten, die Allele eines Kindes zu auszutauschen und damit auch seine Symmetrie zu ändern.

Vorbereitung Das Ziel ist eine Formel für die Wahrscheinlichkeit zu finden, dass der Fitnesswert des ersten Nachkommens um m höher liegt als der des ersten Eltern. Betrachtet werden die Eltern x und y . Ihre Bitstellen werden folgendermaßen bezeichnet:

$$\begin{aligned}x &= x_1^1 x_2^1 \dots x_{l/2}^1 \quad x_{l/2}^2 \dots x_2^2 x_1^2 \\y &= y_1^1 y_2^1 \dots y_{l/2}^1 \quad y_{l/2}^2 \dots y_2^2 y_1^2\end{aligned}$$

Die Rekombinationsmaske M sei

$$M = m_1^1 m_2^1 \dots m_{l/2}^1 \quad m_{l/2}^2 \dots m_2^2 m_1^2.$$

Es wird die Auswirkung der Rekombination an einem Bit x_i^1 und seinem entsprechenden Gegenstück x_i^2 auf der anderen Zeichenkettenhälfte untersucht sowie die Auswirkung für y_i^1 und y_i^2 . Das Element x hat einen Fitnesswert von s_x und y einen Fitnesswert von s_y . Es gibt folgende Fälle:

4 Die Symmetrie als Fitnessfunktion und die gerichtete Rekombination

	Fitnessänderung von x_i^1 und x_i^2	Fitnessänderung von y_i^1 und y_i^2
Fall 1	$x_i^1 = y_i^1$ und $x_i^2 = y_i^2$ 0	0
	Egal, welche Maske verwendet wird, der Fitnessbeitrag dieses Bits bleibt unverändert, da die Genome der Kinder mit denen der Eltern identisch sind (siehe Abbildung 4.3). Die Bezeichnung $P(s_x + m)$ steht für die Wahrscheinlichkeit, dass das erste Kind nach der Rekombination einen Fitnesswert von $s_x + m$ besitzt, $P(s_y + m)$ dagegen bezieht sich auf den zweiten Nachkommen. $P(s_x \pm 0) = P(s_y \pm 0) = 1$	
Fall 2	$x_i^1 \neq y_i^1$ und $x_i^2 = y_i^2$	
	2 a) $x_i^1 = x_1^1$, das heißt x_i ist symmetrisch und y_i ist unsymmetrisch (siehe Abbildung 4.3)	
	$m_i^1 = m_i^2 = 0$ 0	0
	$m_i^1 = m_i^2 = 1$ -1	1
	$m_i^1 = 1, m_i^2 = 0$ -1	1
	$m_i^1 = 0, m_i^2 = 1$ 0	0
	Die Wahrscheinlichkeit, dass das Maskenbit eine bestimmte Ausprägung hat, ist $\frac{1}{2}$, also gilt $P(s_x - 1) = P(s_y + 1) = \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2}$ $P(s_x \pm 0) = P(s_y \pm 0) = \frac{1}{2}$	
	2 b) $x_i^1 \neq x_1^1$, das heißt x_i ist unsymmetrisch und y_i ist symmetrisch	
	$m_i^1 = m_i^2 = 0$ 0	0
	$m_i^1 = m_i^2 = 1$ 1	-1
	$m_i^1 = 1, m_i^2 = 0$ 1	-1
	$m_i^1 = 0, m_i^2 = 1$ 0	0
	$P(s_x + 1) = P(s_y - 1) = \frac{1}{2}$ $P(s_x \pm 0) = P(s_y \pm 0) = \frac{1}{2}$ Für $x_i^1 = y_i^1$ und $x_i^2 \neq y_i^2$ erhält man die gleichen Wahrscheinlichkeiten	
Fall 3	$x_i^1 \neq y_i^1$ und $x_i^2 \neq y_i^2$	
	3 a) $x_i^1 = x_1^1$, also sind x_i und y_i symmetrisch (siehe Abbildung 4.3)	
	$m_i^1 = m_i^2 = 0$ 0	0
	$m_i^1 = m_i^2 = 1$ 0	0
	$m_i^1 = 1, m_i^2 = 0$ -1	-1
	$m_i^1 = 0, m_i^2 = 1$ -1	-1
	$P(s_x - 1) = P(s_y - 1) = \frac{1}{2}$ $P(s_x \pm 0) = P(s_y \pm 0) = \frac{1}{2}$	

4.3 Wirkung der Rekombination in der Fitnesslandschaft

	Fall 1	Fall 2a)	Fall 3a)
1. Elter $x^1 x^2$			
2. Elter $y^1 y^2$			
<hr/>			
$m_1 = m_2 = 0$			
1. Kind			
2. Kind			
$m_1 = m_2 = 1$			
$m_1 = 1, m_2 = 0$			
$m_1 = 0, m_2 = 1$			

Abbildung 4.3: Veranschaulichung der Rekombinationsmöglichkeiten und der Ergebnisse für die Fälle 1, 2a und 3a

	Fitnessänderung des ersten Kindes	Fitnessänderung des zweiten Kindes
3 b) $x_i^1 \neq x_i^2$, das heißt x_i und y_i sind unsymmetrisch		
$m_i^1 = m_i^2 = 0$	0	0
$m_i^1 = m_i^2 = 1$	0	0
$m_i^1 = 1, m_i^2 = 0$	1	1
$m_i^1 = 0, m_i^2 = 1$	1	1
$P(s_x + 1) = P(s_y + 1) = \frac{1}{2}$		
$P(s_x \pm 0) = P(s_y \pm 0) = \frac{1}{2}$		

Fitnessveränderung des ersten Kindes Zunächst wird nur die Fitnessveränderung des ersten Kindes im Vergleich mit dem ersten Elter betrachtet. Später werden die Fitnessveränderungen beider Nachkommen untersucht.

Die Bits einer Zeichenkettenhälfte können wie folgt klassifiziert werden:

N_0 ist die Anzahl an Bits, die unter keinen Umständen zu einer Symmetrieänderung beitragen können (Fall 1)

N_+^x ist die Anzahl an Bits, an denen durch einen Allel-Austausch ihr Fitnessbeitrag um 1 für das erste Kind erhöht werden kann aber nicht muss (Fall 2 b und 3 b)

N_-^x ist entsprechend die Anzahl an Bits, die potentiell zu einer Symmetrieverringeringung beitragen können (Fall 2 a und 3 a)

Es gilt:

$$N_0 + N_+^x + N_-^x = \frac{l}{2}.$$

4 Die Symmetrie als Fitnessfunktion und die gerichtete Rekombination

Handelt es sich zum Beispiel bei den Eltern um

$$\begin{aligned} x &= 0010\ 1010 \\ y &= 0111\ 0011, \end{aligned}$$

so sind

$$\begin{aligned} N_0 &= 1 && \text{(das dritte Bit)} \\ N_+^x &= 2 && \text{(das zweite und das vierte Bit)} \\ N_-^x &= 1 && \text{(das erste Bit)}. \end{aligned}$$

Soll der Fitnesswert des ersten Nachkommens um $m \in \{0, \dots, \frac{l}{2}\}$ höher sein als der des ersten Elters, müssen m Allele von N_+^x ausgetauscht werden. An allen anderen Positionen sollte sich der Fitnessbeitrag nicht ändern. Dabei kann eine positive, das heißt fitnesserhöhende Wirkung der Rekombination durch eine negative wieder aufgehoben werden. Die Anzahl der Fälle, bei denen eine solche Neutralisation eintritt, wird mit n bezeichnet. Man erhält für die Wahrscheinlichkeit, dass sich die Symmetrie des ersten Kindes im Vergleich mit dem ersten Elter um m erhöht bzw. verringert:

$$P(s_x + m) = \left(\frac{1}{2}\right)^{\frac{l}{2} - N_0} \sum_{2n \leq \frac{l}{2} - N_0 - m} \binom{N_+^x}{m+n} \binom{N_-^x}{n} \quad (4.3)$$

$$P(s_x - m) = \left(\frac{1}{2}\right)^{\frac{l}{2} - N_0} \sum_{2n \leq \frac{l}{2} - N_0 - m} \binom{N_+^x}{n} \binom{N_-^x}{m+n}. \quad (4.4)$$

Die Summe läuft über alle möglichen Fälle, bei denen eine Symmetrienerhöhung durch eine -verringering neutralisiert wird. Der Faktor $(\frac{1}{2})^{\frac{l}{2} - N_0}$ beschreibt die Wahrscheinlichkeit, dass die gewünschte Fitnesserhöhung tatsächlich eintritt (siehe Übersicht auf Seite 46).

Um die Verteilung der Fitnessveränderungen des zweiten Kindes bezüglich des zweiten Elters zu berechnen, müssen nur die entsprechenden Werte für N_+^y und N_-^y für das zweite Elter gebildet und in (4.3) und (4.4) eingesetzt werden.

Als Beispiel werden die Eltern $x = 0011$ mit $s_x = 0$ und $y = 1101$ mit $s_y = 1$ betrachtet. Folgende Resultate können bei der Rekombination eintreten:

4.3 Wirkung der Rekombination in der Fitnesslandschaft

Maske	1. Kind	Fitness- änderung	2. Kind	Fitness- änderung
0000	0011	0	1101	0
0001	0011	0	1101	0
0010	0001	1	1111	1
0100	0111	1	1001	1
1000	1011	1	0101	-1
0011	0001	1	1111	1
0101	0111	1	1001	1
1001	1011	1	0101	-1
0110	0101	0	1011	0
1010	1001	2	0111	0
1100	1111	2	0001	0
0111	0101	0	1011	0
1011	1001	2	0111	0
1101	1111	2	0001	0
1110	1101	1	0011	-1
1111	1101	1	0011	-1

Man erhält für das Beispiel folgende Wahrscheinlichkeiten. Es wird später überprüft werden, ob die Wahrscheinlichkeiten des Beispiels mit den im weiteren Verlauf des Abschnitts gefundenen Formeln übereinstimmen:

$$P(s_x \pm 0) = \frac{1}{4}$$

$$P(s_x + 1, s_y - 1) = \frac{1}{4}$$

$$P(\text{Wenigstens ein Nachkomme hat Fitness 1}) = 1.$$

Es ist $N_+^x = 2$, alle weiteren Parameter sind 0. Mit (4.3) oder (4.4) erhält man:

$$P(s_x \pm 0) = \left(\frac{1}{2}\right)^2 \sum_{2n \leq 2} \binom{2}{n} \binom{0}{n} = \frac{1}{4}.$$

Dies steht im Einklang mit dem Beispiel.

Fitnessveränderung beider Kinder Die Fitnessveränderungen der aus der Rekombination hervorgehenden Kinder sind voneinander abhängig. In der Übersicht auf Seite 46 konnte man sehen, dass im zweiten Fall jede Veränderung beim ersten Kind eine komplementäre Wirkung beim zweiten Kind hervorruft. Im dritten Fall jedoch finden die Erhöhungen bzw. die Verringerungen der Fitness im gleichen Maß bei beiden Nachkommen statt.

Neue Notationen müssen eingeführt werden:

4 Die Symmetrie als Fitnessfunktion und die gerichtete Rekombination

- N_{++} ist die Anzahl an Bitstellen, die für beide Nachkommen einen positiven Fitnessbeitrag liefern können, aber nicht müssen
- N_{--} ist die Anzahl an Positionen, die für beide Nachkommen unsymmetrisch werden können
- N_0 ist die Anzahl der Stellen, an denen die Rekombination keinerlei Auswirkung hat
- N_{+-} Anzahl der Bits, an denen eine Erhöhung der Symmetrie des ersten Nachkommens zu Lasten des zweiten Nachkommen geht
- N_{-+} ist dementsprechend die Anzahl an Stellen, an denen die Symmetrie des ersten Nachkommens verringert werden kann und die des zweiten dabei erhöht wird.

Auch hier gilt, dass die Summe $\frac{l}{2}$ ergeben muss:

$$N_{++} + N_{--} + N_0 + N_{+-} + N_{-+} = \frac{l}{2}$$

und es gilt:

$$\begin{aligned} N_+^x &= N_{++} + N_{+-} \\ N_-^x &= N_{--} + N_{-+} \\ N_+^y &= N_{++} + N_{-+} \\ N_-^y &= N_{--} + N_{+-} . \end{aligned}$$

In dem obigen Beispiel ($x=0011$, $y=1101$) sind

$$\begin{aligned} N_{++} &= 1 \\ N_{+-} &= 1, \end{aligned}$$

und alle weiteren Parameter 0.

Allmählich nähert sich das Ziel: Die Formel für die Wahrscheinlichkeit, dass wenigstens ein Nachkomme eine bestimmte Symmetrie von m besitzt. Es wird angenommen, dass dazu die Symmetrie des ersten Nachkommens um m_1 verändert werden müsste oder die Symmetrie des zweiten Nachkommens um m_2 . Es gilt also:

$$\begin{aligned} m &= s_x + m_1 \\ &= s_y + m_2. \end{aligned}$$

Diesmal können m_1 und m_2 auch negative Werte annehmen. Wie viele Fitnessveränderungen tatsächlich an den Positionen von N_{-+} , N_{+-} , N_{++} und N_{--} stattfinden, soll mit n_1 , n_2 , n_3 und n_4 bezeichnet werden. Es muss gelten:

$$\begin{aligned} -n_1 + n_2 + n_3 - n_4 &= m_1 \\ n_1 - n_2 + n_3 - n_4 &= m_2. \end{aligned}$$

4.3 Wirkung der Rekombination in der Fitnesslandschaft

Damit hat man für die Wahrscheinlichkeit, dass beide Kinder einen Symmetriewert von m besitzen:

$$P(s_x + m_1, s_y + m_2) = \left(\frac{1}{2}\right)^{\frac{l}{2}-N_0} \sum_{\substack{-n_1+n_2+n_3-n_4=m_1 \\ n_1-n_2+n_3-n_4=m_2}} \binom{N_{-+}}{n_1} \binom{N_{+-}}{n_2} \binom{N_{++}}{n_3} \binom{N_{--}}{n_4} \quad (4.5)$$

und

$$P(\text{Wenigstens ein Nachkomme hat Symmetrie } m) = P(s_x + m_1) + P(s_y + m_2) - P(s_x + m_1, s_y + m_2). \quad (4.6)$$

Ist zum Beispiel $m_1, m_2 > 0$, so lautet die Gleichung:

$$P(\text{Wenigstens ein Nachkomme hat Symmetrie } m) = \left(\frac{1}{2}\right)^{\frac{l}{2}-N_0} \left(\sum_{2n \leq \frac{l}{2}-N_0-m_1} \binom{N_+^x}{m_1+n} \binom{N_-^x}{n} + \sum_{2n \leq \frac{l}{2}-N_0-m_2} \binom{N_+^y}{m_2+n} \binom{N_-^y}{n} - \sum_{\substack{-n_1+n_2+n_3-n_4=m_1 \\ n_1-n_2+n_3-n_4=m_2}} \binom{N_{-+}}{n_1} \binom{N_{+-}}{n_2} \binom{N_{++}}{n_3} \binom{N_{--}}{n_4} \right)$$

Wendet man sich noch kurz dem obigen Beispiel zu, so findet man:

$$P(s_x + 1, s_y - 1) = \left(\frac{1}{2}\right)^2 \sum_{\substack{n_2+n_3=1 \\ -n_2+n_3=-1}} \binom{0}{n_1} \binom{1}{n_2} \binom{1}{n_3} \binom{0}{n_4} = \frac{1}{4}$$

sowie

$$\begin{aligned} P(\text{Wenigstens ein Nachkomme hat Fitness } 1) &= \\ P(s_x + 1) + P(s_y \pm 0) - P(s_x + 1, s_y \pm 0) &= \\ \frac{1}{2} + \frac{1}{2} - \left(\frac{1}{2}\right)^2 \sum_{\substack{n_2+n_3=1 \\ -n_2+n_3=0}} \binom{0}{n_1} \binom{1}{n_2} \binom{1}{n_3} \binom{0}{n_4} &= \\ \frac{1}{2} + \frac{1}{2} - 0 &= 1. \end{aligned}$$

Diese Wahrscheinlichkeiten stimmen mit den relativen Häufigkeiten im Beispiel überein.

Interpretation Die Variable N_0 zählt die Bits, bei denen die Eltern auf beiden Zeichenkettenhälften übereinstimmen. Je homogener die Population wird, also je mehr Genotypen übereinstimmen, umso höher sollte N_0 pro Elternpaar sein. Mit N_0 kann also ein Maß für die Homogenität der Population geschaffen werden:

$$\frac{1}{(2n)^2} \sum_{I, J \in P_t} \frac{N_0(I, J)}{\frac{l}{2}},$$

wobei $2n$ die Populationsgröße ist und $N_0(I, J)$ die Anzahl an Bits, an deren beiden Stellen I und J übereinstimmen. Dieses Maß ist genau dann 1, wenn die Population vollkommen homogen ist, das heißt wenn sie aus nur einem Genotyp besteht.

Der Faktor $(\frac{1}{2})^{\frac{l}{2}-N_0}$ in (4.6) wird also im Laufe der Homogenisierung immer größer, wohingegen mit wachsenden N_0 die Variablen N_{++} , N_{+-} , N_{-+} und N_{--} kleiner werden und damit auch die Binomialkoeffizienten in der Summe kleiner werden. Damit werden große Fitnessänderungen unwahrscheinlicher.

Nun wurden alle Vorbereitungen getätigt, die für den nächsten Abschnitt benötigt werden. Es wurde die Fitnesslandschaft untersucht und die Auswirkungen der genetischen Operatoren Mutation und Rekombination in der Landschaft formalisiert. Dieses Wissen kann nun angewendet werden, um die Evolutionsgleichung speziell für die Symmetrie als Fitnessfunktion aufzustellen.

4.4 Evolutionsgleichung für die neue Fitnessfunktion

Es ist angebracht, eine neue Äquivalenzklasse einzuführen. Bisher wurde der Suchraum in Schemen unterteilt. Nun sollen aber alle diejenigen Elemente zu einer Klasse zusammengefasst werden, die die gleiche Symmetrie besitzen. Nimmt man also zwei Elemente I und J mit den Symmetrien oder Fitnesswerten s_I und s_J , dann soll gelten:

$$I \sim J \Leftrightarrow s_I = s_J$$

Die Menge aller Elemente aus dem Suchraum \mathfrak{A} , die an s Bitstellen symmetrisch sind, wird mit H_s bezeichnet:

$$H_s = \{I \in \mathfrak{A} : F(I) = s\}.$$

Es gibt $\frac{l}{2} + 1$ Äquivalenzklassen, da s die Werte von 0 bis $\frac{l}{2}$ annehmen kann.

Die Evolutionsgleichung lautet dann:

$$P(H_s, t + 1) = \sum_{k=0}^{\frac{l}{2}} P_m(H_k \rightarrow H_s) P_c(H_k, t). \quad (4.7)$$

4.5 Der gerichtete Rekombinationsoperator

Dabei ist $P_m(H_k \rightarrow H_s)$ die Wahrscheinlichkeit, dass ein Element aus H_k zu einem aus H_s mutiert, das heißt die Fitness von $I \in H_k$ um $s - k$ verändert wird. Die Wahrscheinlichkeit, dass die Fitness nach einer bestimmten Anzahl von Mutationen um einen bestimmten Wert verändert wurde ist durch (4.2) gegeben. Damit hat man:

$$P_m(H_k \rightarrow H_s) = \sum_{l=0}^m p_m^l (1 - p_m)^{l-m} P(k + (s - k) \text{ bei } m \text{ Mutationen}).$$

Der zweite Term $P_c(H_k, t)$ von (4.7) beschreibt die Wahrscheinlichkeit, mit der ein Vertreter von H_k aus der Rekombination hervorgeht. Dies ist gegeben durch:

$$\begin{aligned} P_c(H_k, t) &= (1 - p_c)P'(H_k) + p_c \sum_M \sum_{I,J} p(M)P'(I)P'(J) \cdot \\ &\quad P(\text{wenigstens ein Nachkomme hat Fitness } k) \\ &= (1 - p_c)P'(H_k) + p_c \sum_M \sum_{I,J} p(M)P'(I)P'(J) \left(P(s_I + (k - s_I)) + \right. \\ &\quad \left. P(s_J + (k - s_J)) - P(s_I + (k - s_I), s_J + (k - s_J)) \right). \end{aligned}$$

Für ein bestimmtes Element I bezeichnet der Term $P'(I)$ die Wahrscheinlichkeit zur Rekombination ausgewählt zu werden. Das Argument t wird der Übersichtlichkeit halber weggelassen. Damit wird gesetzt:

$$P'(H_k) = \sum_{I \in H_k} P'(I).$$

Die Wahrscheinlichkeit $P(s_I + (k - s_I))$ ist durch (4.3) bzw. (4.4) gegeben und die von $P(s_I + (k - s_I), s_J + (k - s_J))$ durch (4.5).

Mit diesen Änderungen kann man die Evolutionsgleichung an die neue Fitnessfunktion anpassen. In ihrer Struktur konnte sie durch die Einführung einer neuen Äquivalenzrelation erhalten bleiben. Die Äquivalenzklassen vereinen jetzt Elemente gleicher Symmetrien.

4.5 Der gerichtete Rekombinationsoperator

In diesem Abschnitt soll der Erfolg der Rekombination durch die Einführung eines neuen Rekombinationsoperators erhöht werden. Bisher tritt folgendes Phänomen auf: Kreuzt man zwei stark symmetrische Elemente miteinander, die sehr unterschiedliche Genotypen haben, zum Beispiel 0000 und 1111, so ist die Fitness der Nachkommen mit hoher Sicherheit niedriger als die der Eltern, obwohl die Eltern fitnessmaximal waren.

Die willkürliche Elternwahl ist von Nachteil, wenn stark symmetrische Eltern gekreuzt werden, die sich im Genotyp unterscheiden. Werden jedoch

4 Die Symmetrie als Fitnessfunktion und die gerichtete Rekombination

eher unsymmetrische Eltern gekreuzt, so kann die willkürliche Elternwahl nur von Vorteil sein.

Eine Methode, um dieses Phänomen weitgehend zu vermeiden und die Fitness der Kinder in ähnlicher Wertigkeit wie der Eltern zu belassen, ist es, nur relativ ähnliche Elemente zur Rekombination zuzulassen. Die Wahrscheinlichkeit, mit der sich zwei Elemente kreuzen, soll also mit der Anzahl übereinstimmender Bits steigen. Zum Beispiel kann die Wahrscheinlichkeit, mit der sich zwei Elemente $x = x_1 \dots x_l$ und $y = y_1 \dots y_l$, die zur Rekombination ausgewählt wurden, kreuzen, folgende Funktion sein:

$$\begin{aligned} P_{xy} &:= P(x \text{ und } y \text{ kreuzen sich, falls sie ausgewählt wurden}) \\ &= \frac{1}{l} \sum_{i=1}^l ((1 + x_i + y_i) \bmod 2). \end{aligned}$$

Die Summe zählt die Anzahl an übereinstimmenden Bits, denn es gilt:

$$(1 + x_i + y_i) \bmod 2 = \begin{cases} 0 & x_i \neq y_i \\ 1 & x_i = y_i \end{cases}.$$

Ein Beispiel hierfür ist Folgendes: Findet zwischen x und y eine Rekombination statt, so ist P_{xy} für verschiedene Partner y dann:

x	y	P_{xy}
0000	0000	$4/4 = 1$
	0001	$3/4$
	0011	$2/4$
	0111	$1/4$
	1111	0

Es ist möglich P_{xy} mit $q \in \mathbb{N}$ zu potenzieren, um für $q > 1$ die Differenz von P_{xy_1} und P_{xy_2} zu skalieren (siehe Abbildung 4.4). Und zwar würde die Differenz bei Paaren, die sehr ähnlich sind, steigen und bei eher unähnlichen sinken. Für das Beispiel heißt dies für $q = 4$:

x	y	P_{xy}	P_{xy}^q
0000	0000	1	1
	0001	0.75	0.32
	0011	0.5	0.06
	0111	0.25	0.004
	1111	0	0

Die Wahrscheinlichkeit, dass x und y gekreuzt werden, sinkt damit auf jeden Fall, solange x und y sich an wenigstens einem Bit unterscheiden. Diese Wahrscheinlichkeit fällt umso schneller, je ähnlicher sie sich sind. Die Variable P_{xy}^q ist ein Maß des Abstandes von x und y . Je geringer ihre Hamming-Distanz, umso höher ist P_{xy}^q . Sind x und y identisch, so ist $P_{xy}^q = 1$.

4.5 Der gerichtete Rekombinationsoperator

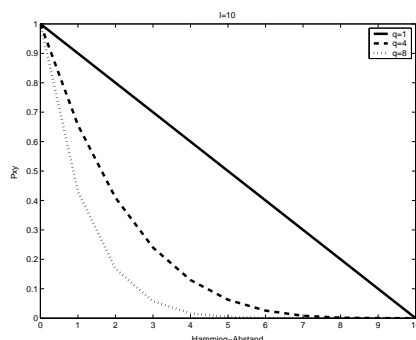


Abbildung 4.4: P_{xy}^q in Abhängigkeit vom Hamming-Abstand

Dieser Operator, bei dem Partner mit gewisser Ähnlichkeit bevorzugt werden, wird *gerichteter Rekombinationsoperator* genannt.

Die Einführung dieses neuen Rekombinationsoperators wird durch folgende Beobachtung gestützt. In der Natur zeigen die meisten Arten eine Vorliebe für Partner, die in bestimmten Eigenschaften ähnlich zu ihnen sind. Eine Auswahl wird jedoch meistens nach dem Phänotyp und eher indirekt nach dem Genotyp getroffen. So suchen sich zum Beispiel Seepferdchen bevorzugt gleichgroße Partner und bestimmte Vögel gleichfarbige Partner.

In der englischsprachigen Literatur nennt man eine solche Partnerpräferenz „assortative mating“, zu Deutsch „assortative Paarung“ oder auch „gerichtete, auswählende Paarung“, und es wird seit geraumer Zeit untersucht, inwieweit diese diskriminierende Auswahl einen Einfluss auf die Entstehung von neuen Arten hat. Die Aufspaltung einer Art in zwei oder mehrere neue Arten durch räumliche Trennung, zum Beispiel durch Kontinentaldrift oder Naturkatastrophen, und anschließende getrennte Weiterentwicklung ist häufig untersucht worden und scheint belegt. Doch das Aufspalten ohne Isolation ist nicht ausreichend untersucht. Es wird in Erwägung gezogen, dass die assortative Partnerwahl zur Artenentstehung ohne räumliche Isolation beigetragen hat. Doch der genaue Einfluss ist unklar und es ist noch nicht geklärt, inwieweit noch weitere Faktoren zusätzlich zur gerichteten Partnerwahl eine Rolle spielen, wie zum Beispiel eine „aufspaltende Selektion“, das heißt eine Selektion in Richtung von zwei neuen Anpassungsoptima.

Nun wird die Wirkung des gerichteten Rekombinationsoperators in der Fitnesslandschaft untersucht. Zu Beginn ist es interessant, die Wahrscheinlichkeit $P_c(I)$, mit der eine Zeichenkette I aus der Rekombination hervorgeht, an den neuen Operator anzupassen. Das Zeitargument wird wieder weggelassen. Bisher galt für die Rekombination (3.19):

$$P_c(I) = (1 - p_c)P'(I) + p_c \sum_M \sum_{J,K} p(M) \lambda_I^{JK}(M) P'(J) P'(K) \quad (4.8)$$

Der erste Term $(1 - p_c)P'(I)$ gibt die Wahrscheinlichkeit wieder, mit der kei-

4 Die Symmetrie als Fitnessfunktion und die gerichtete Rekombination

ne Rekombination stattfindet und das Element unverändert in die nächste Population übernommen wird.

Eine Rekombination kann aus zweierlei Gründen nicht stattfinden. Zum einen, weil sie mit der Wahrscheinlichkeit $1 - p_c$ nicht eintritt, und zum anderen, weil sich zwei zur Rekombination zugelassene Elemente mit einer Wahrscheinlichkeit von $1 - P_{xy}^q$ nicht paaren. Der Unterschied zwischen diesen beiden Fällen ist, dass im ersteren Fall das Element in die Folgepopulation übernommen wird, im zweiten Fall jedoch aussortiert wird.

So wird die Wahrscheinlichkeit $P_{gc}(I)$, mit der eine Zeichenkette I nach der gerichteten Rekombination vorhanden ist, berechnet, indem zu der Wahrscheinlichkeit, dass I unverändert in die nächste Population übernommen wird, die Wahrscheinlichkeit addiert wird, mit der die Zeichenkette I aus einer Kreuzung hervorgeht. Man erhält dann für $P_{gc}(I)$:

$$P_{gc}(I) = (1 - p_c)P'(I) + p_c \sum_M \sum_{J,K} p(M) \lambda_I^{JK}(M) P'(J) P'(K) P_{JK}^q.$$

Man stellt fest, dass $P_{gc}(I)$ im Vergleich mit $P_c(I)$ durch den zusätzlichen Faktor in der letzten Summe verringert wird. Das heißt also, dass sich weniger Individuen kreuzen. Die Rekombinationshäufigkeit wird also gesenkt.

Normierung Summiert man die Evolutionsgleichung über alle Zeichenketten I , so muss das Ergebnis 1 sein. Dies ist mit der Gleichung für den normalen Rekombinationsoperator gegeben, denn

$$\begin{aligned} \sum_I P(I, t+1) &= \sum_I (1 - p_c)P'(I) + \\ &\quad \sum_I p_c \sum_M p(M) \sum_{J,K} \lambda_I^{JK}(M) P'(J) P'(K) \\ &= 1 - p_c + p_c \sum_M p(M) \sum_{J,K} \left(\sum_I \lambda_I^{JK}(M) \right) P'(J) P'(K) \\ &= 1 - p_c + p_c \sum_M p(M) \sum_J P'(J) \sum_K P'(K) \\ &= 1 - p_c + p_c \\ &= 1. \end{aligned}$$

Die Summe $\sum_I \lambda_I^{JK}(M)$ ergibt 1, da zwei bestimmte Eltern und eine bestimmte Maske den ersten Nachkommen eindeutig festlegen. Es kann nur ein eindeutig bestimmtes Individuum als erster Nachkomme resultieren.

Die Evolutionsgleichung für die gerichtete Rekombination muss noch normiert werden, da der zweite Term von (4.8) mit P_{JK}^q gewichtet wurde. Für den Normierungsfaktor Z muss gelten:

4.6 Evolutionsgleichung für die neue Fitnessfunktion und die gerichtete Rekombination

$$\begin{aligned}
 1 &\stackrel{!}{=} Z \sum_{M,I,J,K} p(M) \lambda_I^{JK}(M) P'(J) P'(K) P_{JK}^q \\
 &= Z \sum_{J,K} P'(J) P'(K) P_{JK}^q \\
 \Rightarrow Z &= \frac{1}{\sum_{J,K} P'(J) P'(K) P_{JK}^q}
 \end{aligned}$$

Also erhält man die Evolutionsgleichung mit dem gerichteten Rekombinationsoperator:

$$P(I, t + 1) = \sum_J P_m(J \rightarrow I) P_{gc}(J) \quad (4.9)$$

mit

$$\begin{aligned}
 P_m(J \rightarrow I) &= p_m^{d_{IJ}^H} (1 - p_m)^{l - d_{IJ}^H} \\
 P_{gc}(I) &= (1 - p_c) P'(I) \\
 &\quad + Z p_c \sum_M \sum_{J,K} p(M) \lambda_I^{JK}(M) P'(J) P'(K) P_{JK}^q.
 \end{aligned}$$

Es bezeichnet d_{IJ}^H die Hamming Distanz zwischen den Zeichenketten I und J .

4.6 Evolutionsgleichung für die neue Fitnessfunktion und die gerichtete Rekombination

Um das Kapitel abzurunden, fehlt die Evolutionsgleichung für die Symmetrie-Funktion *und* den gerichteten Rekombinationsoperator. Mit der eingeführten Äquivalenzklasse H_s erhält man sofort:

$$P(H_s, t + 1) = \sum_{k=0}^{\frac{l}{2}} P_m(H_k \rightarrow H_s) P_{gc}(H_k, t) \quad (4.10)$$

mit

$$\begin{aligned}
 P_m(H_k \rightarrow H_s) &= \sum_{l=0}^m p_m^l (1 - p_m)^{l-m} P(k + (s - k) \text{ bei } m \text{ Mutationen}) \\
 P_c(H_k, t) &= (1 - p_c) P'(H_k) + Z p_c \sum_M \sum_{I,J} p(M) P'(I) P'(J) P_{JK}^q \cdot \\
 &\quad P(\text{wenigstens ein Nachkomme hat Fitness } k).
 \end{aligned}$$

Homogenisierung der Population Die Angleichung des Erbguts wird weiterhin durch das Zusammenspiel von Selektion und Rekombination erreicht, auch mit dem gerichteten Rekombinationsoperator. Das Aussortieren von isolierten Elementen geht nun noch schneller vonstatten, da die Ähnlichkeit eines Elementes zu einem anderen bei der Rekombination eine Rolle spielt. Befindet sich ein isoliertes Element I in der Population, so ist P_{IJ}^q für alle Partner J gering, da die Hamming-Distanz von I zu allen J größer als Null und auf Grund der Isolation auch größer als die durchschnittliche Hamming-Distanz der Populationselemente ist.

Als Konsequenz müsste die Endpopulation eines Genetischen Algorithmus mit gerichteter Rekombination homogener sein als bei Verwendung der normalen Rekombination. Ob dies tatsächlich der Fall ist, wird in Kapitel 7 beantwortet.

4.7 Zusammenfassung

In diesem Kapitel wurde eine neue Fitnessfunktion, die Spiegelsymmetrie, und ein neuer Rekombinationsoperator, die gerichtete Rekombination, eingeführt. Die Wirkung der Mutation sowie der beiden Rekombinationsarten auf der Fitnesslandschaft wurde neu analysiert. Die Untersuchungen mündeten in einer Evolutionsgleichung, die an die neue Fitnessfunktion und den neuen Rekombinationsoperator angepasst ist (4.10).

5 Effektive Fitness

In diesem Kapitel wird das Konzept der Effektiven Fitness vorgestellt. Theorie und Resultate sind in [SS02], [Ste98] und [Ste99] veröffentlicht.

Die Bewertung der Elemente durch eine Fitnessfunktion ist für die meisten Selektionsoperatoren essentiell. Die Fitnessfunktion bietet aber auch eine Möglichkeit die Evolution in einem Genetischen Algorithmus intuitiv als Suche nach dem Fitnessmaximum zu verstehen. Daher wurde versucht, die Arbeit eines genetischen Algorithmus durch eine Hill-Climbing-Methode auf einer Fitnesslandschaft zu erklären. Die Fitnesslandschaft kann man sich bildlich als den Graphen über dem Suchraum vorstellen, der durch die Fitnessfunktion erzeugt wird.

Obwohl Fitnesslandschaften beim Verständnis der Populationsentwicklung helfen könnten und auch eine Klassifikation von Landschaften wünschenswert wäre, sind die Fortschritte in der Landschaftsanalyse gering. Zum einen ist sie oftmals wegen der Komplexität der Fitnessfunktion oder des schwierig zu definierenden Abstandsbegriffes im Suchraum sehr kompliziert und daher auf einfache Modelle, insbesondere Modelle mit statischer, das heißt zeitunabhängiger, Fitnessfunktion, beschränkt. Zum anderen gibt es keinen sichtbaren Zusammenhang zwischen der Anzahl der lokalen Maxima einer Landschaft und der Schwierigkeit für einen Genetischen Algorithmus das Problem zu lösen. So ist zum Beispiel die zerklüftete Landschaft, die durch die „Stachelschweinfunktion“ erzeugt wird, wesentlich komplizierter als die Landschaft der „Nadel im Heuhaufen“-Funktion. Für einen Genetischen Algorithmus stellt das letztere Problem jedoch die größere Herausforderung dar.

Weitere Unstimmigkeiten beim Betrachten des Genetischen Algorithmus als Hill-Climbing-Prozess tauchen mit dem Genotyp-Phänotyp-Symmetriebruch und der sogenannten Orthogenese auf. Orthogenese ist das Phänomen, bei dem ein Element geringerer Fitness dem fitnessmaximalen Element vorgezogen wird. Dies ist zum Beispiel der Fall, wenn die Fitnesslandschaft zwei voneinander isolierte Maxima besitzt, wobei das kleinere von einem Plateau von relativ fitten Elementen umgeben ist. Der Rest des Suchraumes besteht aus Elementen mit verschwindender Fitness. Siehe dazu Abbildung 5.1. Das Fitnessmaximum ist mit f_m bezeichnet, die zweitgrößte Fitness mit f_s und die Fitness der Plateauelemente mit f_p . Mit diesem Beispiel kann gezeigt werden, dass mit steigender Mutationsrate der größere Teil der Endpopulation aus Elementen mit der zweitgrößten Fitness f_s besteht [Ste99].

Als Genotyp-Phänotyp-Symmetrie bezeichnet man die Äquivalenz unterschiedlicher Genotypen, das heißt unterschiedlicher Zeichenketten, mit demgleichen Phänotyp, das heißt gleichem Fitnesswert. Zu Beginn würde

5 Effektive Fitness

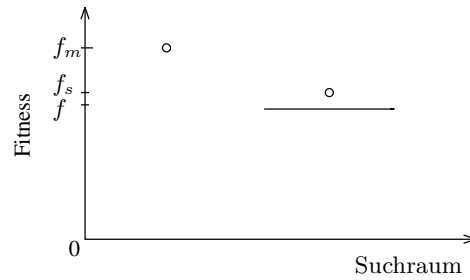


Abbildung 5.1: Plateaufitnessfunktion

man vermuten, dass Elemente mit gleichem Fitnesswert gleichberechtigt behandelt würden und in der Endpopulation mit demselben Anteil vertreten wären. In der Praxis ist dies aber nur selten der Fall. Einfachstes Beispiel hierfür ist das „Nadel im Heuhaufen“-Modell, in dem in der Endpopulation, abgesehen vom Maximum, solche Genotypen dominieren, die eine geringere Hamming-Distanz zum Maximum besitzen, als der Durchschnitt [SS02].

Da die Landschaftsanalyse eine hilfreiche Methode in der Theorie der Genetischen Algorithmen sein kann, überdachte man die Fitness als einzige mögliche Bewertungsfunktion und führte ein neues Fitnesskonzept ein: die *effektive Fitness*. Vorrangiges Ziel war eine genauere Information über den Fortpflanzungserfolg der einzelnen Elemente. Der tatsächliche Fortpflanzungserfolg hängt immer auch von dem Zustand der Population ab und somit auch von der Zeit, sowie von den Einflüssen aller genetischen Operatoren. Der Nachteil der ursprünglichen Fitness war, dass sie ausschließlich Aussagen über den Selektionserfolg der Elemente gab und zudem zeitunabhängig war. Nach [SS02] wird die effektive Fitness f_{eff} für ein Element I implizit definiert:

$$P(I, t + 1) = \frac{f_{eff}(I, t)}{\bar{F}_t} P(I, t). \quad (5.1)$$

Der Nachteil ist, dass die effektive Fitness somit nur für Populationselemente definiert ist.

Die Formel (5.1) erinnert stark an die Wirkung des fitnessproportionalen Selektionsoperators, die ähnlich aussieht:

$$P'(I, t) = \frac{F(I)}{\bar{F}_t} P(I, t) = 2n f_t(I) P(I, t).$$

An dieser Stelle kann man sehen, dass die effektive Fitness allein durch ihre Definition als Proportionalitätsfaktor einen genaueren Schluss auf den Fortpflanzungserfolg liefert, denn in (5.1) sind die Effekte aller genetischen Operatoren einbezogen, also auch die der auf den Genotypen agierenden Operatoren Mutation und Rekombination. Schon hier wird klar, dass die Genotyp-Phänotyp-Symmetrie durch die effektive Fitness nicht automatisch erhalten bleibt.

Mit Hilfe der Evolutionsgleichung für $P(I, t+1)$ (3.19) kann die effektive Fitness folgendermaßen quantifiziert werden:

$$\begin{aligned} f_{eff}(I, t) &= \frac{\bar{F}_t}{P(I, t)} P(I, t+1) \\ &= \frac{\bar{F}_t}{P(I, t)} \sum_J P_m(J \rightarrow I) P_c(J, t) \end{aligned} \quad (5.2)$$

Setzt man in (5.2) $p_m = p_c = 0$, also verzichtet man auf die Rekombination und Mutation:

$$\begin{aligned} P_m(J \rightarrow I) &= \begin{cases} 1 & J = I \\ 0 & J \neq I \end{cases} \\ P_c(J, t) &= P'(J, t) \\ &= 2n f_t(J) P(J, t), \end{aligned}$$

dann gilt bei Verwendung der fitnessproportionalen Selektion $f_{eff}(I, t) = F(I)$. Doch auch wenn die Selektion nicht oder nicht ausschließlich auf der normalen Fitness agiert, sieht man die realistischere Bewertung des Selektionserfolges eines Elementes durch die effektive Fitness.

Das Konzept der effektiven Fitness hat weitere Vorteile abgesehen von der bereits genannten Funktion als Maß für den Fortpflanzungserfolg und somit für die genauere Beschreibung der Endpopulation. Die effektive Fitness kann auch als Maß für den Symmetriebruch dienen, als Erklärung der Orthogenese, als Maß für die Fehlleitung („deception“), sowie als Maß für die Entfernung der Population vom Gleichgewichtszustand. Das Genannte soll im Folgenden genauer erklärt werden, begonnen wird mit einem Beispiel, das die Messbarkeit des Genotyp-Phänotyp-Symmetriebruchs durch die effektive Fitness belegt.

Genotyp-Phänotyp-Symmetriebruch Betrachtet wird das bereits erwähnte „Nadel im Heuhaufen“-Problem [SS02]. Es gibt ein ausgezeichnetes Element mit der höchsten Fitness f_{ms} , dabei steht ms für „master species“. Alle weiteren Elemente haben die geringere Fitness f . Die Rekombination wird außer Acht gelassen ($p_c = 0$), die fitnessproportionale Selektion verwendet und eine homogen verteilte Population angenommen. Das heißt alle Elemente sind mit dem gleichen Anteil in der Population enthalten. Bei einer Populationsgröße von $2n$ gilt also:

$$P(I, t) = \frac{1}{2n} \quad \forall I \in P_t.$$

5 Effektive Fitness

Man erhält:

$$\begin{aligned}
 f_{eff}(I, t) &= \frac{\bar{F}_t}{P(I, t)} \sum_J P_m(J \rightarrow I) 2n f_t(J) P(J, t) \\
 &= \sum_J P_m(J \rightarrow I) F(J), \quad \text{da } f_t(J) = \frac{F(J)}{2n\bar{F}_t} \\
 \Rightarrow f_{eff}(I, t) &= f (1 - p_m)^l + f \sum_{\substack{J \neq ms \\ J \neq I}} p_m^{d_{I,J}^H} (1 - p_m)^{l - d_{I,J}^H} \\
 &\quad + f_{ms} p_m^{d_{I,ms}^H} (1 - p_m)^{l - d_{I,ms}^H}
 \end{aligned}$$

$$\begin{aligned}
 \Delta f_{eff}(I_1, I_2) &= f_{eff}(I_1, t) - f_{eff}(I_2, t) \\
 &= f_{ms} \left(p_m^{d_{I_1,ms}^H} (1 - p_m)^{l - d_{I_1,ms}^H} \right. \\
 &\quad \left. - p_m^{d_{I_2,ms}^H} (1 - p_m)^{l - d_{I_2,ms}^H} \right) \\
 &\quad + f \left(- p_m^{d_{I_1,ms}^H} (1 - p_m)^{l - d_{I_1,ms}^H} \right. \\
 &\quad \left. + p_m^{d_{I_2,ms}^H} (1 - p_m)^{l - d_{I_2,ms}^H} \right) \\
 &= (f_{ms} - f) \left(p_m^{d_{I_1,ms}^H} (1 - p_m)^{l - d_{I_1,ms}^H} \right. \\
 &\quad \left. - p_m^{d_{I_2,ms}^H} (1 - p_m)^{l - d_{I_2,ms}^H} \right)
 \end{aligned}$$

Für $\Delta f_{eff}(I_1, I_2)$ heben sich die Terme der Summe $\sum_{\substack{J \neq ms \\ J \neq I}}$ bis auf die beiden Terme auf, in denen

$$\begin{aligned}
 d_{I_1, J}^H &= d_{I_2, ms}^H \\
 d_{I_2, J}^H &= d_{I_1, ms}^H
 \end{aligned}$$

gilt. Es wird angenommen, dass $d_{I_1, ms}^H < d_{I_2, ms}^H$. Also soll I_1 näher an ms liegen als I_2 . Dann folgt

$$\begin{aligned}
 \Delta f_{eff}(I_1, I_2) &= (f_{ms} - f) p_m^{d_{I_1,ms}^H} (1 - p_m)^{l - d_{I_1,ms}^H} \\
 &\quad \cdot \left(1 - \left(\frac{p_m}{1 - p_m} \right)^{d_{I_2,ms}^H - d_{I_1,ms}^H} \right) \tag{5.3}
 \end{aligned}$$

Für $p_m < \frac{1}{2}$ wird die Differenz der effektiven Fitnesswerte zweier Elemente mit steigendem Abstand $d_{I_2,ms}^H - d_{I_1,ms}^H$ immer größer. Das heißt der

effektive Fitnessvorteil von I_1 gegenüber I_2 steigt mit wachsenden Abstand zu I_2 . Somit kann man den Genotyp-Phänotyp-Symmetriebruch zwischen I_1 und I_2 , die dieselben Fitnesswerte besitzen, mit Hilfe des Fortpflanzungsvorteils Δf_{eff} messen. An diesem Beispiel kann man erkennen, dass die effektive Fitness auch die neutrale Evolution erklären kann. Hat man ein flaches Gebiet in der Fitnesslandschaft, so wird es in der Richtung verlassen, in der der positive Fitnessgradient am größten ist.

Orthogenese Es wird nun die Orthogenese betrachtet, das Phänomen, bei dem ein submaximales Element gegenüber dem maximalen Element bevorzugt wird [Ste98], [Ste99], zum Beispiel wenn das submaximale von einem Plateau von relativ fitten Elementen umgeben wird und die Mutationsrate eine bestimmte Schwelle übersteigt. Das Fitnessmaximum wird mit m , das submaximale Element mit s bezeichnet. Die Fitnesswerte sind mit f_m und f_s gekennzeichnet. Die Plateauelemente p haben die Fitness f_p , alle anderen Elemente verschwindende Fitness. Es wird mit $p_c = 0$ gerechnet, mit fitnessproportionaler Selektion und homogen verteilter Population. Dann erhält man:

$$\begin{aligned}
f_{eff}(m) &= f_m (1 - p_m)^l \\
&\quad + f_s p_m^{d_{m,s}^H} (1 - p_m)^{l - d_{m,s}^H} \\
&\quad + f_p \sum_{I_p} p_m^{d_{m,I_p}^H} (1 - p_m)^{l - d_{m,I_p}^H} \\
f_{eff}(s) &= f_s (1 - p_m)^l \\
&\quad + f_m p_m^{d_{m,s}^H} (1 - p_m)^{l - d_{m,s}^H} \\
&\quad + f_p \sum_{I_p} p_m^{d_{s,I_p}^H} (1 - p_m)^{l - d_{s,I_p}^H} \\
\Delta f_{eff}(m, s) &= (f_m - f_s)(1 - p_m)^l \\
&\quad + (f_s - f_m) p_m^{d_{m,s}^H} (1 - p_m)^{l - d_{m,s}^H} \\
&\quad + f_p \sum_{I_p} \left(p_m^{d_{m,I_p}^H} (1 - p_m)^{l - d_{m,I_p}^H} \right. \\
&\quad \quad \left. - p_m^{d_{s,I_p}^H} (1 - p_m)^{l - d_{s,I_p}^H} \right) \\
&= (f_m - f_s) \left((1 - p_m)^l - p_m^{d_{m,s}^H} (1 - p_m)^{l - d_{m,s}^H} \right) \\
&\quad - f_p \sum_{I_p} p_m^{d_{s,I_p}^H} (1 - p_m)^{l - d_{s,I_p}^H} \\
&\quad \cdot \left(1 - \left(\frac{p_m}{1 - p_m} \right)^{d_{m,I_p}^H - d_{s,I_p}^H} \right)
\end{aligned}$$

5 Effektive Fitness

Für $p_m < \frac{1}{2}$ ist der zweite Term von $\Delta f_{eff}(m, s)$ positiv und der dritte negativ. Man erkennt, dass der effektive Fitnessvorteil von m durch die Plateauelemente gedämpft wird.

Fehlleitung Einem weiteren Bereich der Theorie über Genetische Algorithmen, nämlich den der Fehlleitung („deception“), kann das Konzept der effektiven Fitness zu Fortschritten verhelfen. Im Allgemeinen betrachtet man ein Problem als fehlleitend, wenn Schemenbausteine den Genetischen Algorithmus vom globalen Fitnessmaximum wegführen, das heißt die Bausteine eines Schemas besitzen eine höhere Fitness als das Gesamtschema. Mit der Möglichkeit den tatsächlichen Fortpflanzungserfolg eines Schemas zu messen, kann man die Differenz zwischen effektiver Fitness und normaler Fitness als Maß für die Fehlleitung betrachten [SW99]. An dieser Stelle wird nochmals auf das zuletzt betrachtete Beispiel verwiesen, bei dem das submaximale Element von einem Plateau relativ fitter Elemente umgeben ist und das globale Maximum isoliert ist. Auch dieses Problem betrachtet man als vermeintlich fehlleitend, da das submaximale Element durchaus bevorzugt werden konnte. Es wird klar, dass das globale Maximum der Fitnessfunktion eines Problems nicht automatisch mit dem Optimum unter dem Einfluss von genetischen Operatoren übereinstimmt. Als Konsequenz für die Praxis müsste ein Genetischer Algorithmus derart an das Problem angepasst werden, dass das Fitnessmaximum und das Optimum der Evolution zusammenfallen.

Entfernung vom Gleichgewichtszustand Die effektive Fitness fungiert auch als Maß für die Entfernung der Population vom Gleichgewichtszustand [Ste99]. Setzt man die Gleichgewichtsbedingung $P(I, t + 1) = P(I, t)$ in (5.1) ein, so folgt

$$f_{eff}(I, t) = \bar{F}_t \quad \forall I \in P_t \quad \forall t.$$

Das heißt also, dass die effektive Fitnesslandschaft im Gleichgewichtszustand flach ist. Ein Beispiel soll diese zusätzliche Maßeigenschaft veranschaulichen. Betrachtet wird eine flache Fitnesslandschaft sowie eine homogen verteilte Population, das heißt alle Elemente sind mit gleicher Häufigkeit vertreten. Die Rekombination wird der Einfachheit halber wieder vernachlässigt sowie die fitnessproportionale Selektion verwendet. Somit ist auch die effektive Fitnesslandschaft flach. Um das zu erkennen, braucht man nur in (5.3) $f = f_{ms}$ setzen, da alle Elemente die gleiche Fitness haben. Weicht aber die Population vom homogenen Zustand ab, so spiegelt sich dies in einer nicht flachen effektiven Fitnesslandschaft wider. Betrachtet wird der Suchraum $\{00, 11, 01, 10\}$ und die Population $\{11, 11, 11, 01, 01, 10, 10, 00\}$. Man berechnet mit $f = 1$:

$$\begin{aligned}
f_{eff}(I) &= \frac{1}{P(I,t)} \sum_J P_m(J \rightarrow I) P(J,t) \\
\Rightarrow f_{eff}(11) &= (1-p_m)^2 + \frac{8}{3} \left(\frac{1}{2} p_m (1-p_m) + \frac{1}{8} p_m^2 \right) \\
&= 1 - \frac{2}{3} p_m \\
f_{eff}(01) &= (1-p_m)^2 + 4 \left(\frac{1}{2} p_m (1-p_m) + \frac{1}{4} p_m^2 \right) \\
&= 1 \\
&= f_{eff}(10) \\
f_{eff}(00) &= (1-p_m)^2 + 8 \left(\frac{1}{2} p_m (1-p_m) + \frac{3}{8} p_m^2 \right) \\
&= 1 + 2p_m.
\end{aligned}$$

Die effektive Fitness von 00 ist also größer als die aller anderen Elemente und 11 hat die niedrigste effektive Fitness. Es ist zu bemerken, dass der Fortpflanzungserfolg von 01 und 10 unabhängig von der Mutationsrate ist. Die Elemente 01 und 10 haben bereits ihre Gleichgewichtsverteilung in der Population erreicht.

Nähert sich die Population dem homogenen Zustand an, so wird dies in der Flachheit der effektiven Fitnesslandschaft sichtbar. Numerische Beispiele für die Konvergenz der effektiven Fitnesslandschaft an eine flache Landschaft, während sich die Population homogen ordnet, findet man in [Ste99].

In diesem Kapitel konnte man sehen, wie das eingeführte Konzept der effektiven Fitnesslandschaft bisherige Unklarheiten beseitigt und die Theorie der Genetischen Algorithmen vereinfacht. Die Ausführungen basieren auf [Ste98], [Ste99], [SW99] und [SS02].

5 *Effektive Fitness*

6 Basen des Suchraums

Während sich die vorherigen Kapitel eher auf den Algorithmus konzentrierten sowie die Verbreitung von Schemen und konzeptionelle Betrachtungen, soll nun der Suchraum in den Vordergrund gerückt werden. Das Ziel ist es, den Suchraum als Vektorraum zu beschreiben und einige Basen für ihn und seinen Dualraum aufzustellen. So eignen sich einige Basen für die Beschreibung der genetischen Operatoren besser als andere. Eine Basis wird die Anwendung einer Methode aus der Statistischen Mechanik ermöglichen, mit der die Dynamik der Evolution im Genetischen Algorithmus behandelt werden kann.

Dieses Kapitel basiert hauptsächlich auf den Resultaten von [CS]. In jener Veröffentlichung wurde auf Beweise verzichtet. Sämtliche Beweise in diesem Kapitel sind daher für diese Arbeit hergeleitet wurden.

6.1 Der Suchraum und sein Dualraum

Nach Definition arbeitet ein Genetischer Algorithmus mit binären Zeichenketten. Somit besteht der Suchraum \mathfrak{A} aus der Menge aller binären Zeichenketten der Länge l :

$$\mathfrak{A} = \underbrace{\{0, 1\} \times \dots \times \{0, 1\}}_{l\text{-mal}} = \{0, 1\}^l.$$

Geometrisch kann man sich den Suchraum als die Eckpunkte des l -dimensionalen Einheitswürfel vorstellen. Wie in Abbildung 6.1 zu sehen, ist zum Beispiel für den 3-dimensionalen Fall die Zeichenkette 101 durch den Eckpunkt $(1,0,1)$ repräsentiert.

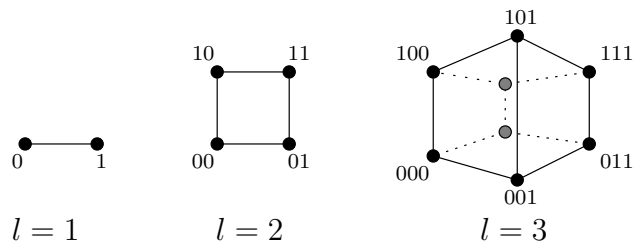


Abbildung 6.1: Der l -dimensionale Einheitswürfel

So kann \mathfrak{A} in den \mathbb{R}^l eingebettet werden. Dies macht an dieser Stelle aber keinen besonderen Sinn, da \mathfrak{A} dann auch die additive Struktur des \mathbb{R}^l erben würde. Die Addition von binären Zeichenketten mag zwar noch

6 Basen des Suchraums

nützlich sein, jedoch haben die Elemente des Suchraums für jedes Optimierungsproblem eine andere Bedeutung. Sie sind Kodierungen von Lösungen und eine solche Lösung muss nicht mit der Zahl, deren Binärdarstellung mit der Kodierung identisch ist, übereinstimmen. So ist die Addition von Individuen nur im Ausnahmefall sinnvoll. Daher sollte sich ein Individuum im Allgemeinen nicht als Linearkombination von anderen Individuen darstellen lassen. Für die Anschauung ist die Einbettung in den \mathbb{R}^l zwar hilfreich, es eignet sich für unsere Zwecke jedoch mehr, den Suchraum in den $\mathbb{R}^{|\mathfrak{A}|} = \mathbb{R}^{2^l} = \mathbb{R}^N$ mit $N := 2^l$ einzubetten. Der Suchraum \mathfrak{A} ist die Familie aller binären Zeichenketten der Länge l :

$$\mathfrak{A} = \{x_1, \dots, x_N\},$$

dann sollte die Einbettungsvorschrift ϕ

$$\begin{aligned} \phi : \mathfrak{A} &\rightarrow \mathbb{R}^N \\ x_i &\mapsto e_i \quad i = 1, \dots, N \end{aligned}$$

so erfolgen, dass die Menge $\{e_i\}_{i=1}^N$ linear unabhängig ist.

Der \mathbb{R}^N ist bekanntlich ein Vektorraum. Auch im Fall der Einbettung von \mathfrak{A} in den \mathbb{R}^N erbt der Suchraum die additive Struktur. Nun braucht man sich jedoch keine Sorgen mehr zu machen, dass ein Individuum als Summe von anderen Individuen darstellbar ist. Hierfür hat man mit der Forderung nach linearer Unabhängigkeit bei der Einbettungsvorschrift gesorgt. Werden dennoch Individuen addiert, so liegt das Ergebnis außerhalb von \mathfrak{A} .

Der Raum der reellwertigen Funktionen über \mathfrak{A}

$$Fun(\mathfrak{A}) = \{f : \mathfrak{A} \rightarrow \mathbb{R}\}.$$

ist mit der üblichen punktweisen Addition und der skalaren Multiplikation ein Vektorraum. Die beiden Räume \mathbb{R}^N und $Fun(\mathfrak{A})$ sind isomorph. Der nächstliegende Isomorphismus ψ ist:

$$\begin{aligned} \psi : Fun(\mathfrak{A}) &\rightarrow \mathbb{R}^N \\ f &\mapsto (f(x_1), \dots, f(x_N)). \end{aligned}$$

Es bezeichne $\{f_i\}_{i=1}^N$ die kanonische Basis von $Fun(\mathfrak{A})$:

$$f_i(x_j) = \delta_{ij},$$

dabei ist δ_{ij} das Kronecker Delta. Die i -te Basisfunktion ist also nur an dem i -ten Element gleich 1, sonst 0. Die Basis entspricht unter ψ der kanonischen Basis im \mathbb{R}^N , und die Basiselemente können als charakteristische Funktionen zu den einzelnen Elementen von \mathfrak{A} betrachtet werden.

Die Einführung des Funktionenraums $Fun(\mathfrak{A})$ ist insofern sinnvoll, als dass später damit die Evolutionsgleichungen verkürzt werden können. Die Population kann als N -elementiger Vektor aufgefasst werden, wobei die i -te Komponente die relative Häufigkeit des i -ten Elementes von \mathfrak{A} wiedergibt.

Der Dualraum von $Fun(\mathfrak{A})$ ist:

$$Fun^*(\mathfrak{A}) = \{g : Fun(\mathfrak{A}) \rightarrow \mathbb{R}, g \text{ ist linear}\}.$$

Die Dimension von $Fun^*(\mathfrak{A})$ ist N . Somit ist auch $Fun^*(\mathfrak{A})$ zu \mathbb{R}^N isomorph.

Für die duale Basis $\{g_i\}_{i=1}^N$ zu $\{f_i\}_{i=1}^N$ muss gelten:

$$g_i(f_j) = \delta_{ij}.$$

So folgt für ein beliebiges Element $f := \psi(f) = (\alpha_1, \dots, \alpha_N) \in Fun(\mathfrak{A})$ und ein beliebiges Element $g := \psi(g) = (\beta_1, \dots, \beta_N) \in Fun^*(\mathfrak{A})$:

$$\begin{aligned} g(f) &= \beta_1 g_1(f) + \dots + \beta_N g_N(f) \\ &= \beta_1 g_1(\alpha_1 f_1 + \dots + \alpha_N f_N) + \dots + \beta_N g_N(\alpha_1 f_1 + \dots + \alpha_N f_N) \\ &= \alpha_1 \beta_1 + \dots + \alpha_N \beta_N, \end{aligned}$$

denn die Basisfunktion g_i gibt den Wert von f am i -ten Element wieder:

$$g_i(f) = \alpha_i.$$

Somit kann man ein Element $g \in Fun^*(\mathfrak{A})$ als Linearkombination der Funktionswerte an den Elementen von \mathfrak{A} auffassen.

Mit der Abbildung

$$\begin{aligned} \nu : \mathfrak{A} &\rightarrow Fun^*(\mathfrak{A}) \\ x_i &\mapsto g_i \end{aligned}$$

kann \mathfrak{A} in $Fun^*(\mathfrak{A})$ eingebettet werden. Auch hier ist die Bedingung, dass die Bilder der x_i linear unabhängig sind, erfüllt, da sie injektiv auf die Basiselemente abgebildet werden.

Bisher wurde gezeigt, in welche Struktur sich die Menge \mathfrak{A} einbetten lässt. In den folgenden Abschnitten werden einige Basen für $Fun(\mathfrak{A})$ konstruiert. Die Evolutionsgleichung soll mit Hilfe der verschiedenen Basen ausgedrückt werden und es wird ersichtlich, dass jeder genetische Operator mit einer anderen Basis am besten beschrieben werden kann.

6.2 Basen von $Fun(\mathfrak{A})$

Es werden drei Basen eingeführt, die bei der theoretischen Behandlung von Genetischen Algorithmen eine Rolle spielen. Dabei handelt es sich um die Stringbasis, die Bausteinbasis und die Walsh-Basis.

Stringbasis Die Stringbasis ist die kanonische Basis. Sie wurde mit $\{f_i\}_{i=1}^N$ bereits hergeleitet und besteht aus den charakteristischen Funktionen der einelementigen Teilmengen von \mathfrak{A} . Zur Abgrenzung von den anderen Basen wird für sie das Symbol B_S vergeben:

$$B_S = \{f_i\}_{i=1}^N.$$

6 Basen des Suchraums

Zeitweise wird in dieser Arbeit die charakteristische Funktion zu einem Element auch mit dem Element selbst indiziert:

$$B_S := \{f_{x_i}\}_{i=1}^N.$$

Building block basis, Bausteinbasen Eine Bausteinbasis wird bezüglich eines Elementes x gebildet. Dazu wird die Menge der Schemen, die x beinhalten, betrachtet. Diese Menge ist der B_x -Träger:

$$B_x\text{-Träger} = \bigcup_{\substack{s \in \Xi \\ x \in s}} s.$$

Für ein Schema s des B_x -Trägers sind seine Ordnung und die Anzahl der mit x übereinstimmenden Bits identisch. Das Element x wird auch als Bezugspunkt bezeichnet.

Als Beispiel für den Bezugspunkt 110 ist der B_x -Träger die Menge $\{***, **0, *1*, *10, 1** , 1*0, 11*, 110\}$.

Die Bausteinbasis zu x , oder auch B_x , wird dann gebildet aus der Menge der charakteristischen Funktionen, deren Träger ein x enthaltenes Schema ist. Der Träger einer Basisfunktion ist also im B_x -Träger enthalten:

$$B_x = \bigcup_{\substack{s \in \Xi \\ \text{mit} \\ x \in s}} \{f_s\} = \bigcup_{s \in B_x\text{-Träger}} \{f_s\}.$$

Die charakteristischen Funktionen werden mit ihren Trägern indiziert. Das Bezugselement x ist in allen Schemen, die als Indexwerte verwendet werden, enthalten und umgekehrt ist jedes Schema, welches x beinhaltet, als Indexwert vorhanden.

Da zu jedem $x \in \mathfrak{A}$ eine andere Bausteinbasis konstruiert werden kann, gibt es N verschiedene Bausteinbasen. Die Elemente einer Bausteinbasis nennt man Bausteine oder Building blocks.

Der B_x -Träger kann mit B_x identifiziert werden, denn es gibt eine Bijektion zwischen der Potenzmenge des Suchraums $\mathfrak{P}(\mathfrak{A})$ und der Menge der charakteristischen Funktionen über \mathfrak{A} . Jede Teilmenge $B \subset \mathfrak{A}$ definiert eine charakteristische Funktion, nämlich diejenige, deren Träger mit B identisch ist. Und umgekehrt repräsentiert jede charakteristische Funktion eine Teilmenge von \mathfrak{A} , und zwar diejenige, die mit dem Träger identisch ist. Es besteht also eine Bijektion zwischen der Potenzmenge von \mathfrak{A} und der Menge der charakteristischen Funktionen über \mathfrak{A} und man kann die beiden Mengen miteinander identifizieren. Damit kann auch B_x mit dem B_x -Träger identifiziert werden.

Abbildung 6.2 verdeutlicht den Unterschied zwischen dem Schemenraum und einer Bausteinbasis. Der Anschaulichkeit halber wird \mathfrak{A} hier wieder in den \mathbb{R}^l eingebettet. Man sieht, dass ein k -Schema geometrisch eine $(l - k)$ -dimensionale Hyperfläche des \mathfrak{A} repräsentiert. Im obigen Beispiel für drei Dimensionen ist ein 3-Schema also ein Punkt des Suchraums, ein 2-Schema

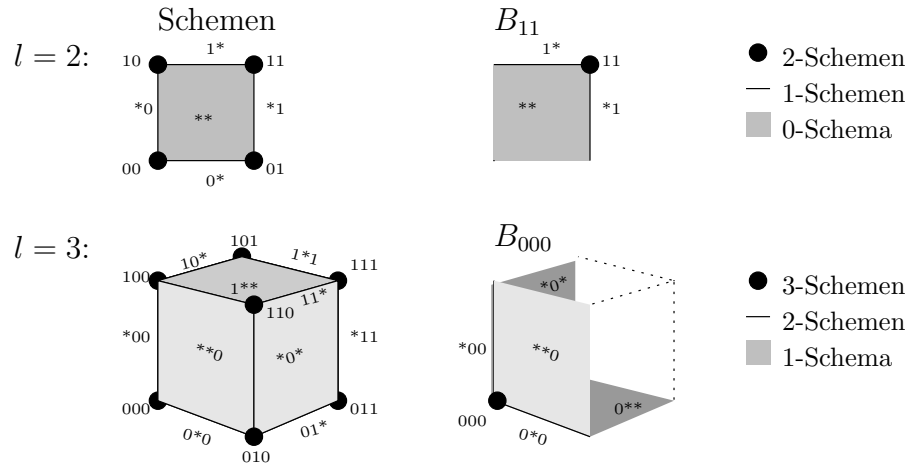


Abbildung 6.2: Geometrische Veranschaulichung von Schemen und Bausteinen

repräsentiert eine Kante des Würfels, ein 1-Schema eine Seite und das 0-Schema $**$ den gesamten Würfel. Jeder Punkt des Suchraums ist also in je $\binom{l}{k}$ k -Schemen enthalten, hier also in einem 0-Schema, in drei 1-Schemen, in drei 2-Schemen und natürlich in einem 3-Schema.

Nachweis der Basiseigenschaften von B_x Die Elemente der Stringbasis, also die charakteristischen Funktionen der Elemente von \mathfrak{A} , sollen als Linearkombination der Elemente von B_x dargestellt werden. Damit keine Verwechslungen zwischen dem Bezugspunkt x und den sonstigen Elementen von \mathfrak{A} auftreten, wird je nach Notwendigkeit ein Element von \mathfrak{A} auch mit c bezeichnet. Unter Verwendung der aus der Kombinatorik bekannten Siebformel findet man:

$$f_c = f_s + \sum_{k=1}^{l-o(s)} (-1)^k \sum_{\substack{f_{s^k} \in B_x \\ s^k \subset s \\ o(s^k)=k+o(s)}} f_{s^k} \quad \forall c \in \mathfrak{A}. \quad (6.1)$$

Dabei ist $s = (s_1, \dots, s_l)$ das Schema aus dem B_x -Träger mit der größten Ordnung, in dem x und c enthalten sind:

$$s_j = \begin{cases} x_j, & \text{falls } c_j = x_j \\ *, & \text{sonst} \end{cases} \quad j = 1, \dots, l,$$

wobei nach Konstruktion von s die Ordnung, $o(s)$, die Anzahl der übereinstimmenden Bits von x und c ist. Es gilt $f_s \in B_x$, da $x \in s$.

Die Summationsbedingung sieht zwar verwirrend aus, bedeutet jedoch nichts anderes, als dass die Summe über all diejenigen Teilschemen s^k von s läuft, die im B_x -Träger enthalten sind und eine um k höhere Ordnung als s haben. Damit haben diese Schemen k Bits mehr mit x gemeinsam und sind an k Bits mehr von c verschieden als s .

6 Basen des Suchraums

Es wird von der charakteristischen Funktion in B_x ausgegangen, die als Träger das kleinste Schema hat, in dem x und c liegen: f_s . Von dieser Funktion werden all die charakteristischen Funktionen aus B_x abgezogen, deren Träger ein Schema ist, welches an genau einem zusätzlichen Bit mit x statt mit c übereinstimmt. Danach werden die charakteristischen Funktionen dazu addiert, deren Träger mehrfach abgezogen wurden. Anschließend werden wieder die Funktionen subtrahiert, die mehrfach dazu addiert wurden, und so weiter, bis der Träger der rechten Seite nur noch die Menge $\{c\}$ ist.

Bevor der Beweis von Formel (6.1) geliefert wird, werden zur Veranschaulichung zwei Beispiele für $l = 3$ und $l = 4$ betrachtet: Für $l = 3$ und den Bezugspunkt $x = 000$ ist der B_x -Träger gleich

$$\{***, **0, *0*, *00, 0**, 0*0, 00*, 000\}$$

und somit ist

$$B_x = \{f_{***}, f_{**0}, f_{*0*}, f_{*00}, f_{0**}, f_{0*0}, f_{00*}, f_{000}\}.$$

Die charakteristischen Funktionen für die Elemente 001, 011 und 111 können dann mit (6.1) folgendermaßen dargestellt werden:

$$\begin{aligned} f_{001} &= f_{00*} - f_{000} \\ f_{011} &= f_{0**} - (f_{00*} + f_{0*0}) + f_{000} \\ f_{111} &= f_{***} - (f_{0**} + f_{**0} + f_{*0*}) + (f_{00*} + f_{0*0} + f_{*00}) - f_{000}. \end{aligned}$$

Abbildung 6.3 veranschaulicht die Summation. Die Tabelle ist folgendermaßen zu lesen: Die betreffende charakteristische Funktion, die als Linearkombination dargestellt werden soll, ist in der ersten Spalte eingetragen. Die zweite Spalte zeigt den Träger von f_s und die weiteren Spalten beziehen sich auf die alternierende Summe. Die Summanden zu einem k sind untereinander eingetragen. Jedes Bild zeigt den Zwischenwert von f_c . Die Funktionswerte von f_c sind farbig kodiert.

Bei dem vierdimensionalen Fall muss die Visualisierung leider entfallen. Hier ist

$$B_x = \{f_{****}, f_{***0}, f_{**0*}, f_{**00}, f_{*0**}, f_{*0*0}, f_{*00*}, f_{*000}, f_{0***}, f_{0**0}, f_{0*0*}, f_{0*00}, f_{00**}, f_{00*0}, f_{000*}, f_{0000}\}$$

und

$$\begin{aligned} f_{0001} &= f_{000*} - f_{0000} \\ f_{0011} &= f_{00**} - (f_{000*} + f_{00*0}) + f_{0000} \\ f_{0111} &= f_{0***} - (f_{00**} + f_{0*0*} + f_{0**0}) + (f_{000*} + f_{00*0} + f_{0*00}) - f_{0000} \\ f_{1111} &= f_{****} - (f_{000*} + f_{00*0} + f_{0*00} + f_{*000}) \\ &\quad + (f_{00**} + f_{0*0*} + f_{0**0} + f_{*00*} + f_{**00} + f_{*0*0}) \\ &\quad - (f_{000*} + f_{00*0} + f_{0*00} + f_{*000}) \\ &\quad + f_{0000}. \end{aligned}$$

	s	$k = 1$	$k = 2$	$k = 3$
f_{001}				
f_{011}				
f_{111}				

Funktionswerte verschieden von Null sind dargestellt durch:

● $f = 1$ ○ $f = -1$ ○ $f = -2$

Abbildung 6.3: Linearkombination der Elemente von B_{000} für $f_{001}, f_{011}, f_{111}$

Nun folgt der Beweis von (6.1).

Satz 6.1 B_x ist für $\text{Fun}(\mathfrak{A})$ eine Basis und es gilt:

$$f_c = f_s + \sum_{k=1}^{l-o(s)} (-1)^k \sum_{\substack{f_{s^k} \in B_x \\ s^k \subset s \\ o(s^k)=k+o(s)}} f_{s^k} \quad \forall c \in \mathfrak{A}.$$

Beweis:

(i) Es soll zunächst die Formel bewiesen werden. Gilt diese, so ist B_x ein Erzeugendensystem. Zu Beginn wird für die rechte Seite der Formel die Abkürzung \bar{f}_c eingeführt.

Schritt 1: Zu zeigen: $\bar{f}_c(c) = 1$. Es werden die Träger der charakteristischen Funktionen auf der rechten Seite betrachtet. Es gilt $c \in s$ und $c \notin s^k$, denn c ist nur in s , nicht in den s^k , enthalten, da s nach Konstruktion das Schema der höchsten Ordnung unter den Elementen des B_x -Trägers ist,

6 Basen des Suchraums

welches noch c enthält. Daraus folgt $f_s(c) = 1$ und $f_{s^k}(c) = 0$. Demnach gilt: $\bar{f}_c(c) = 1$.

Schritt 2: Zu zeigen: $\bar{f}_c(y) = 0$ für $y \neq c$. Es wird folgender Hilfssatz benötigt.

Hilfssatz 6.2 Jedes s^k ist ein Durchschnitt von k s^1 -Schemen:

$$s^k = \bigcap_{i=1}^k s_i^1 \hat{=} k\text{-facher Durchschnitt der } s^1.$$

Beweis: Es gilt $s^k \subset \dots \subset s^2 \subset s^1 \subset s$. Ein s^k -Schema hat eine Ordnung von $o(s) + k$, besitzt also k fixierte Bits mehr als s . Die zusätzlich fixierten Bits werden mit b_1, \dots, b_k bezeichnet.

Betrachtet man für ein b_i das s^1 -Schema, welches an b_i fixiert ist, so sieht man dass s^k in genau k solcher s^1 -Schemen enthalten ist. Das Schema s^k ist also ein Durchschnitt von k s^1 -Schemen, die genau $o(s) + 1$ Bit mit s^k gemeinsam haben.

Umgekehrt definiert der Durchschnitt von k s^1 -Schemen genau ein s^k -Schema, da der Durchschnitt ein Schema ist, dass zum einen ebenfalls in s liegt, wie auch die s^1 -Schemen, und zum anderen an den b_1, \dots, b_k zusätzlich zu den fixierten Bits von s fixiert ist. Damit ist die Ordnung des Durchschnitt-Schemas $o(s) + k$ und es handelt sich tatsächlich um ein s^k -Schema. \square

Nun werden die Funktionswerte an den Stellen $y \neq c$, $y \in s$ untersucht. Wegen der Inklusion $y \in s$ folgt:

$$\bar{f}_c(y) = 1 - \dots$$

Sei y nun r -mal in den Schemen s^1 enthalten, so wird in der Summation für $k = 1$ von f_s der Wert r abgezogen:

$$\bar{f}_c = 1 - r + \dots$$

Im nächsten Schritt der Summation ($k = 2$) wird die Ordnung von s^1 um 1 erhöht, das heißt, es werden die charakteristischen Funktionen, deren Träger der Schnitt von je zwei Trägern von f_{s^1} ist, wieder dazu addiert. Das Element y ist in $\binom{r}{2}$ Schnitten enthalten. Es gibt $\binom{r}{2}$ f_{s^2} -Funktionen, die addiert werden, und es folgt:

$$\bar{f}_c = 1 - r + \binom{r}{2} - \dots$$

Danach werden die charakteristischen Funktionen mit Trägern entsprechend der 3-er Durchschnitte, das heißt mit Trägern, die den Durchschnitt von je drei Trägern von f_{s^1} repräsentieren, wieder abgezogen. Das Element y ist in $\binom{r}{3}$ von diesen Trägern enthalten:

$$\bar{f}_c = 1 - r + \binom{r}{2} - \binom{r}{3} + \dots$$

Diese Reihe wird bis $k = r$ fortgesetzt. Dabei überschreitet r den Wert $l - o(s)$ nicht, da sich die erste Summation nur über maximal $\binom{l-o(s)}{1} = l - o(s)$ Schemen erstrecken kann. Es gilt also für $y \in s \setminus c$:

$$\begin{aligned}\bar{f}_c(y) &= 1 - r + \binom{r}{2} - \binom{r}{3} + \dots \pm \binom{r}{r-1} \mp \binom{r}{r} \\ &= \sum_{k=0}^r (-1)^k \binom{r}{k}.\end{aligned}$$

1. Fall: Ist r ungerade, so folgt aus der geraden Anzahl der Summanden, da die Summation bei 0 beginnt, der Eigenschaft der Binomialkoeffizienten

$$\binom{r}{k} = \binom{r}{r-k}$$

und der alternierenden Addition, dass

$$\bar{f}(y) = 0 \quad \text{für } y \in s \setminus c.$$

2. Fall: Ist r gerade, so ist die Anzahl an Summanden ungerade. Unter Verwendung einer weiteren Eigenschaft der Binomialkoeffizienten, nämlich

$$\binom{r}{k} = \binom{r-1}{k-1} + \binom{r-1}{k},$$

kann man berechnen:

$$\begin{aligned}& \binom{r}{0} - \binom{r}{1} + \binom{r}{2} - \dots - \binom{r}{r-1} + \binom{r}{r} \\ &= 1 - \left(\binom{r-1}{0} + \binom{r-1}{1} \right) + \left(\binom{r-1}{1} + \binom{r-1}{2} \right) \\ & \quad - \dots - \left(\binom{r-1}{r-2} + \binom{r-1}{r-1} \right) + 1 \\ &= 2 - \binom{r-1}{0} - \binom{r-1}{r-1} \\ &= 0.\end{aligned}$$

Also gilt für alle $y \in s \setminus c$:

$$\bar{f}_c(y) = 0.$$

Für $y \notin s$, also Elemente außerhalb des Trägers von f_s , ist der Funktionswert generell 0.

Somit wurde bewiesen, dass $\bar{f}_c = f_c$ und dass durch (6.1) tatsächlich die Elemente von B_S dargestellt werden. Die Menge B_x ist somit für beliebiges x ein Erzeugendensystem von $\text{Fun}(\mathfrak{A})$.

6 Basen des Suchraums

(ii) Es bleibt zu zeigen, dass B_x ein minimales Erzeugendensystem ist. Dies ist schnell getan, denn bekanntlich haben die Basen eines Vektorraumes alle die gleiche Länge, das heißt die gleiche Anzahl an Basiselementen. Die Länge von B_x stimmt mit der von B_S überein. Beide haben N Elemente. Die Menge B_x ist also tatsächlich eine Basis und zwar unabhängig von der Wahl von x . \square

Walsh-Basis Eine weitere Basis des $Fun(\mathfrak{A})$ ist die Walsh-Basis B_W . Auf diese wird an späterer Stelle näher eingegangen (Abschnitt 6.3). Der interessierte Leser ist zudem auf [Wri99] verwiesen.

6.3 Basiswechsel

Der Satz (6.1) beschreibt letztendlich die Transformation von den Elementen einer Bausteinbasis auf die Elemente der Stringbasis. Diese Beschreibung ist zwar anschaulich, aber umständlich. Jetzt soll die komponentenweise Transformation durch die Vektortransformation ersetzt werden. Es sollen Übergangsmatrizen für die Übergänge von Stringbasis zu einer Bausteinbasis, zur Walsh-Basis und zwischen zwei verschiedenen Bausteinbasen konstruiert werden. Hat man erst einmal die Übergangsmatrizen gefunden, so kann man leicht die Dualbasen für $Fun^*(\mathfrak{A})$ berechnen.

Notationsvereinbarung Für den Beweis der Transformationsregel ist es hilfreich, folgende Notation einzuführen. Der Baustein xb_i mit $x \in \{0, 1, *\}$ bezeichnet denjenigen Baustein, der durch Anfügen von x vor dem Baustein b_i erzeugt wird. Ist b_i ein Baustein der Länge l , so hat xb_i die Länge $l + 1$.

Analog wird mit den Zeichenketten verfahren. Die Zeichenkette xc_i mit $x \in \{0, 1\}$ wird durch Anfügen von x an die Zeichenkette c_i erzeugt.

Auf diese Weise kann schnell eine Basis höherer Dimension gebildet werden. Soll zum Beispiel ausgehend von der Bausteinbasis $B_{x_1x_2}$ die Basis $B_{yx_1x_2}$ mit $y \in \{0, 1\}$ konstruiert werden, bzw. aus der Stringbasis von $l = 2$ die Stringbasis zu $l = 3$, so kann man von den geordneten Vektoren der Basiselemente

$$\begin{aligned} v_B^T &= (f_{**}, f_{*x_2}, f_{x_1*}, f_{x_1x_2}) \quad \text{und} \\ v_S^T &= (f_{\bar{x}_1\bar{x}_2}, f_{\bar{x}_1x_2}, f_{x_1\bar{x}_2}, f_{x_1x_2}) \end{aligned}$$

die neuen Vektoren bilden, indem man zuerst $*$ bzw. $\bar{y} = 1 - y$ an die Indexwerte anfügt und dann y . Also erhält man

$$\begin{aligned} v_B'^T &= (f_{***}, f_{**x_2}, f_{*x_1*}, f_{*x_1x_2}, f_{y**}, f_{y*x_2}, f_{yx_1*}, f_{yx_1x_2}) \\ v_S'^T &= (f_{\bar{y}\bar{x}_1\bar{x}_2}, f_{\bar{y}\bar{x}_1x_2}, f_{\bar{y}x_1\bar{x}_2}, f_{\bar{y}x_1x_2}, f_{y\bar{x}_1\bar{x}_2}, f_{y\bar{x}_1x_2}, f_{yx_1\bar{x}_2}, f_{yx_1x_2}). \end{aligned}$$

Der Vektor, der aus allen f_{*b_j} , $j = 1, \dots, N$ besteht, wird mit \vec{f}_{*b} bezeichnet und man setzt analog:

$$\begin{aligned}\vec{f}_{*b} &:= (f_{*b_1}, \dots, f_{*b_N})^T \\ \vec{f}_{yb} &:= (f_{yb_1}, \dots, f_{yb_N})^T \quad \forall f_{b_i} \in B_{b_N} \\ \vec{f}_{\bar{y}c} &:= (f_{\bar{y}c_1}, \dots, f_{\bar{y}c_N})^T \\ \vec{f}_{yc} &:= (f_{yc_1}, \dots, f_{yc_N})^T \quad \forall f_{c_i} \in B_S.\end{aligned}$$

Auch die Elemente von \mathfrak{A} können analog für die Dimension $l+1$ gebildet werden:

$$\mathfrak{A} = \{\bar{y}x, yx\} = \{\bar{y}x_1, \dots, \bar{y}x_N, yx_1, \dots, yx_N\}.$$

Die geordnete Menge $\bar{y}x$ symbolisiert dann alle diejenigen Elemente des Suchraumes der Dimension $l+1$, die am ersten Bit ein \bar{y} besitzen, die Menge yx dementsprechend die Elemente mit y an der ersten Stelle.

Der Vorteil dieser Notation ist nicht nur die vereinfachte Dimensionserhöhung, sondern auch die Beibehaltung der speziellen Anordnung, die im nächsten Paragraphen vorgestellt wird.

Anordnungsvorschrift der Basenelemente Die Herleitung der Übergangsmatrizen für die Dimension l aus denjenigen für $l-1$ wird vereinfacht, wenn die Tensormultiplikation verwendet wird. Diese kann aber nur angewendet werden, wenn die Basenelemente auf der linken und der rechten Seite eine bestimmte Ordnung besitzen. Es wird sich auf folgende Anordnung geeinigt: Die letzte Vektorkomponente stimmt überein, z. B. 000 in der Bausteinbasis und 000 in der Stringbasis, und von dem letztem Element ausgehend wird zuerst das letzte Bit variiert, also 00* in der Bausteinbasis und 001 in der Stringbasis, dann das zweitletzte (0*0 und 010), dann die beiden letzten (0** und 011), dann das drittletzte (*00 und 100). Dabei ändern sich die hinteren Bits im Folgenden in genau der gleichen Reihenfolge wie zuvor. Die Variation wird bis zum ersten Bit fortgesetzt.

Mit dieser Ordnung gibt es nun auch N verschieden geordnete Stringbasen. Zu jedem Bezugselement gibt es genau eine. Wird im Folgenden eine bestimmte geordnete Stringbasis benötigt, gibt das Bezugselement die Ordnung der Stringbasenelemente an. Es wird angenommen, dass auch die Elemente des Suchraums bezüglich des Bezugspunktes angeordnet sind.

Mit dem Bezugselement $x = x_1 \dots x_l$ sieht der Vektor der Elemente von B_x dann formell so aus:

$$\begin{pmatrix} f_* & \dots & * \\ \vdots & & \\ f_{x_1 \dots x_{l-2}} & * & * \\ f_{x_1 \dots x_{l-2}} & * & x_l \\ f_{x_1} & \dots & x_{l-1} * \\ f_{x_1} & \dots & x_{l-1} x_l \end{pmatrix}$$

6 Basen des Suchraums

und der Vektor der Stringbasiselemente:

$$\begin{pmatrix} f_{\bar{x}_1 \dots \bar{x}_l} \\ \vdots \\ f_{x_1 \dots x_{l-2} \bar{x}_{l-1} \bar{x}_l} \\ f_{x_1 \dots x_{l-2} \bar{x}_{l-1} x_l} \\ f_{x_1 \dots x_{l-1} \bar{x}_l} \\ f_{x_1 \dots x_{l-1} x_l} \end{pmatrix}.$$

Als Beispiel wird die Bausteinbasis und Stringbasis mit $l = 3$ für das Bezugsэлемент 000 betrachtet. Dann sind die Basiselemente folgendermaßen angeordnet:

$$\begin{pmatrix} f_{***} \\ f_{**0} \\ f_{*0*} \\ f_{*00} \\ f_{0**} \\ f_{0*0} \\ f_{00*} \\ f_{000} \end{pmatrix} \quad \text{und} \quad \begin{pmatrix} f_{111} \\ f_{110} \\ f_{101} \\ f_{100} \\ f_{011} \\ f_{010} \\ f_{001} \\ f_{000} \end{pmatrix}.$$

Die Anordnungsvorschrift ist nicht nur für die Tensormultiplikation essentiell. Wegen der Isomorphie zwischen $Fun(\mathfrak{A})$ und \mathbb{R}^N können die Vektoren der Basiselemente als Matrizen M_B^l für die Bausteinbasis und M_S^l für die Stringbasis geschrieben werden:

$$\begin{matrix} & \begin{matrix} 111 & 110 & 101 & 100 & 011 & 010 & 001 & 000 \end{matrix} \\ \begin{matrix} *** \\ **0 \\ *0* \\ *00 \\ 0** \\ 0*0 \\ 00* \\ 000 \end{matrix} & \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ & & 1 & 1 & 0 & 0 & 1 & 1 \\ & & & 1 & 0 & 0 & 0 & 1 \\ & & & & 1 & 1 & 1 & 1 \\ & & & & & 1 & 0 & 1 \\ & & & & & & 1 & 1 \\ & & & & & & & 1 \end{pmatrix} & = M_B^3 \end{matrix} \tag{6.2}$$

$$\begin{matrix} \begin{matrix} 111 \\ 110 \\ 101 \\ 100 \\ 011 \\ 010 \\ 001 \\ 000 \end{matrix} & \begin{pmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{pmatrix} & = M_S^3 \end{matrix}$$

Die Komponenten von M_B^3 und M_S^3 , die weggelassen wurden sind 0. Die Spalten korrespondieren zu den Elementen von \mathfrak{A} und die Zeilen zu den Basiselementen, sie sind daher klein an jede Spalte und Zeile geschrieben. Das heißt konkret, dass ein Matrixelement $(M_B^l)_{ij}$ oder $(M_S^l)_{ij}$ den Funktionswert des j -Elementes des Suchraums an der i -ten Basisfunktion wiedergibt:

$$(M_B^l)_{ij} = f_{b_i}(x_j) \quad \text{und} \quad (M_S^l)_{ij} = f_{x_i}(x_j).$$

Wegen der Isomorphie zwischen $Fun(\mathfrak{A})$ und dem \mathbb{R}^N kann man schreiben:

$$\psi(v_B) = M_B^l \quad \text{und} \quad \psi(v_S) = M_S^l,$$

wobei ψ auf die Komponenten von v_B und v_S angewendet wird.

Wird das Bezugselement von x zu x_{neu} gewechselt und die neuen Basiselemente in $B_{x_{neu}}$ sowie in B_S nach der Anordnungsvorschrift geordnet, so bleiben die Matrizen in dieser einfachen Form bestehen. Die Anordnungsvorschrift ist also notwendig, damit generelle Übergangsmatrizen für den Basiswechsel angegeben werden können, die unabhängig vom Bezugselement sind.

In (6.2) ist folgende Gesetzmäßigkeit zu erkennen:

$$M_B^l = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{\otimes l} \quad \text{und} \quad M_S^l = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}^{\otimes l} = E_2^{\otimes l} = \underbrace{E_2 \otimes \dots \otimes E_2}_{l\text{-mal}} \quad (6.3)$$

Die Matrix E_2 bezeichnet die 2×2 -Einheitsmatrix. Die Tensormultiplikation \otimes ist gegeben durch

$$A \otimes B = \begin{pmatrix} aB & bB \\ cB & dB \end{pmatrix} = \begin{pmatrix} ax & ay & bx & by \\ az & aw & bz & bw \\ cx & cy & dx & dy \\ cz & cw & dz & dw \end{pmatrix}, \quad \text{wenn}$$

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \text{und} \quad B = \begin{pmatrix} x & y \\ z & w \end{pmatrix}.$$

Es wird nun (6.3) bewiesen.

Hilfssatz 6.3 Die Matrizen der Basenelemente der Stringbasis M_S^l und einer Bausteinbasis M_B^l zu beliebigen Bezugspunkt erfüllen folgende Gleichungen, wenn die Basiselemente bezüglich des Bezugspunktes angeordnet werden:

$$M_B^l = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{\otimes l} \quad \text{und} \quad M_S^l = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}^{\otimes l} = E_2^{\otimes l}.$$

6 Basen des Suchraums

Beweis: Begonnen wird mit der Stringbasis. Die i -te Basisfunktion von B_S ist genau dann 1, wenn das Argument das i -te Element von \mathfrak{A} ist. Ist also \mathfrak{A} sowie B_S bezüglich desselben Bezugsselement geordnet, ist M_S notwendigerweise die Einheitsmatrix der Dimension N . Diese Matrix kann auch als Tensorprodukt dargestellt werden:

$$M_S^l = E_N = E_2^{\otimes l}.$$

Nun wird die Matrix M_B^l betrachtet. Die Gesetzmäßigkeit wird induktiv bewiesen. Der Induktionsanfang für $l = 1$ ist mit

$$\begin{pmatrix} f_* \\ f_0 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

beispielhaft für den Bezugspunkt $x = 0$ schon getan. Das Matricelement $(M_B^1)_{ij}$ ist genau dann 1, wenn $x_j \in \mathfrak{A}$ im i -ten Schema b_i des B_x -Trägers enthalten ist, der ebenfalls nach dem Anordnungssystem geordnet ist.

Die Induktionsvoraussetzung ist

$$M_B^l = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{\otimes l}.$$

Im Induktionsschritt $l \rightarrow l + 1$ wird B_{yx} betrachtet. Dabei ist $y \in \{0, 1\}$. Nach der Konvention über die Konstruktion einer Basis der nächsthöheren Dimension erhält man den neuen Vektor der Basenelemente von B_{yx} :

$$\begin{pmatrix} \vec{f}_{*b} \\ \vec{f}_{yb} \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix},$$

wobei die Matrizen A , B , C und D vom Typ $N \times N$ sind. Es sind \vec{f}_{*b} und \vec{f}_{yb} Vektoren der Länge N , die nach dem speziellem Ordnungssystem geordnet sind.

Damit ist $C = 0$, da keines von den Elementen von $\bar{y}x$ in den Schemen yb enthalten ist, denn im ersten Bit sind sie verschieden. Es gilt $D = M_B^l$, denn ein Element yx_j ist genau dann in einem Schema yb_i enthalten, wenn x_j in b_i enthalten war. Mit demselben Argument ist $A = M_B^l$ und $B = M_B^l$. Daher gilt tatsächlich:

$$M_B^{l+1} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \otimes M_B^l = M_B^{\otimes(l+1)}.$$

□

Transformationen zwischen Stringbasis und Bausteinbasis Begonnen wird mit dem einfachsten Fall: $l = 1$.

Die Transformationsmatrix Λ , die die Koordinatentransformation von der Stringbasis auf die Bausteinbasis B_0 bezeichnen soll, muss die folgende Gleichung erfüllen

$$\begin{pmatrix} f_* \\ f_0 \end{pmatrix} = \Lambda \begin{pmatrix} f_1 \\ f_0 \end{pmatrix} \quad \text{bzw.} \quad \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = \Lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Es gilt also:

$$\Lambda = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

Dabei ist λ_{ij} nur dann gleich 1, wenn der Träger des j -ten Elements der Stringbasis im Träger des i -ten Elements der Bausteinbasis liegt:

$$\lambda_{ij} = \begin{cases} 1 & x_j \in b_i \\ 0 & \text{sonst} \end{cases}.$$

Die Inverse ist

$$\Lambda^{-1} = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}.$$

Für die Transformation von der Bausteinbasis auf die Stringbasis hat man also:

$$\begin{pmatrix} f_1 \\ f_0 \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} f_* \\ f_0 \end{pmatrix} = \begin{pmatrix} f_* - f_0 \\ f_0 \end{pmatrix}$$

Dies steht im Einklang mit Satz (6.1).

Betrachtet wird nun der zweidimensionale Fall:

$$\begin{pmatrix} f_{**} \\ f_{*0} \\ f_{0*} \\ f_{00} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f_{11} \\ f_{10} \\ f_{01} \\ f_{00} \end{pmatrix} = (\Lambda \otimes \Lambda) \begin{pmatrix} f_{11} \\ f_{10} \\ f_{01} \\ f_{00} \end{pmatrix}.$$

Die Inverse zu $\Lambda^{\otimes 2}$ ist dann

$$(\Lambda^{\otimes 2})^{-1} = \begin{pmatrix} 1 & -1 & -1 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix} = (\Lambda^{-1})^{\otimes 2}.$$

Dies führt zu folgendem Satz.

6 Basen des Suchraums

Satz 6.4 Die Übergangsmatrix von der Stringbasis auf eine beliebige Bausteinbasis bei entsprechender Anordnung und einer Zeichenkettenlänge von l ist $\Lambda^{\otimes l}$ mit

$$\Lambda = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

bzw. von einer Bausteinbasis auf die Stringbasis

$$\left(\Lambda^{\otimes l}\right)^{-1} = \left(\Lambda^{-1}\right)^{\otimes l} \quad \text{mit} \quad \Lambda^{-1} = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}.$$

Beweis: Es bezeichne Λ^l die Übergangsmatrix von der Stringbasis auf die Bausteinbasis zur Stringlänge l . Daraus folgt sogleich:

$$\begin{aligned} M_B^l &= \Lambda^l M_S^l \\ &= \Lambda^l E_2^{\otimes l} \\ \Rightarrow \Lambda^l &= M_B^l \\ &= \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{\otimes l} \\ &= \Lambda^{\otimes l}. \end{aligned}$$

Es folgt die umgekehrte Richtung: Die Transformation von der Bausteinbasis auf die Stringbasis. Es wird also die Inverse zu $\Lambda^{\otimes l}$ gesucht. Es gilt allgemein:

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}.$$

Daraus folgt sofort

$$\left(\Lambda^{\otimes l}\right)^{-1} = \left(\Lambda^{-1}\right)^{\otimes l} \quad \text{mit} \quad \Lambda^{-1} = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}.$$

□

Transformationen zwischen verschiedenen Bausteinbasen Wie bereits erwähnt wurde, ist die Bausteinbasis keineswegs eindeutig. Es gibt N verschiedene Bausteinbasen, nämlich zu jedem Suchraumelement genau eine. Die Transformation von einer Bausteinbasis auf eine andere soll formalisiert werden. Dazu wird wieder mit dem eindimensionalen Fall begonnen. Die Übergangsmatrix für $l = 1$ wird mit Λ_0^1 bezeichnet:

$$\begin{aligned} \begin{pmatrix} f_* \\ f_0 \end{pmatrix} &\stackrel{!}{=} \Lambda_0^1 \begin{pmatrix} f_* \\ f_1 \end{pmatrix} \quad \text{bzw.} \quad \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = \Lambda_0^1 \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \\ \Rightarrow \Lambda_0^1 &= \begin{pmatrix} 1 & 0 \\ 1 & -1 \end{pmatrix} \quad \text{und} \quad \left(\Lambda_0^1\right)^{-1} = \Lambda_0^1. \end{aligned}$$

Für $l = 2$ findet man, vorausgesetzt man verwendet das bereits erwähnte Ordnungsverfahren:

$$\begin{aligned}
B_{11} &\longrightarrow B_{00} \\
\begin{pmatrix} f_{**} \\ f_{*0} \\ f_{0*} \\ f_{00} \end{pmatrix} &\stackrel{!}{=} \Lambda_{00}^{11} \begin{pmatrix} f_{**} \\ f_{*1} \\ f_{1*} \\ f_{11} \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} f_{**} \\ f_{*1} \\ f_{1*} \\ f_{11} \end{pmatrix} \\
&\implies \Lambda_{00}^{11} = \Lambda_0^1 \otimes \Lambda_0^1 = (\Lambda_0^1)^{\otimes 2}.
\end{aligned}$$

$$\begin{aligned}
B_{11} &\longrightarrow B_{10} \\
\begin{pmatrix} f_{**} \\ f_{*0} \\ f_{1*} \\ f_{10} \end{pmatrix} &\stackrel{!}{=} \Lambda_{11}^{10} \begin{pmatrix} f_{**} \\ f_{*1} \\ f_{1*} \\ f_{11} \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} f_{**} \\ f_{*1} \\ f_{1*} \\ f_{11} \end{pmatrix} \\
&\implies \Lambda_{11}^{10} = E_2 \otimes \Lambda_0^1.
\end{aligned}$$

$$\begin{aligned}
B_{11} &\longrightarrow B_{01} \\
\begin{pmatrix} f_{**} \\ f_{*1} \\ f_{0*} \\ f_{01} \end{pmatrix} &\stackrel{!}{=} \Lambda_{11}^{01} \begin{pmatrix} f_{**} \\ f_{*1} \\ f_{1*} \\ f_{11} \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} f_{**} \\ f_{*1} \\ f_{1*} \\ f_{11} \end{pmatrix} \\
&\implies \Lambda_{11}^{01} = \Lambda_0^1 \otimes E_2.
\end{aligned}$$

Allgemein gilt:

Satz 6.5 *Mit dem speziellen Ordnungsverfahren sieht die Übergangsmatrix*

$$\Lambda_{x_1 x_2 \dots x_l}^{y_1 y_2 \dots y_l}$$

6 Basen des Suchraums

von einer Bausteinbasis $B_{x_1x_2\dots x_l}$ auf eine andere Bausteinbasis $B_{y_1y_2\dots y_l}$ wie folgt aus:

$$\Lambda_{x_1x_2\dots x_l}^{y_1y_2\dots y_l} = \Lambda_1 \otimes \Lambda_2 \otimes \dots \otimes \Lambda_l \text{ mit}$$

$$\Lambda_i = \begin{cases} \Lambda_0^1 & , x_i \neq y_i \\ E_2 & , x_i = y_i \end{cases} \quad i = 1, \dots, l.$$

Beweis: Dieser Beweis wird induktiv geführt. Der Induktionsanfang wurde bereits gezeigt. Die Behauptung im obigen Satz wird als Induktionsvoraussetzung angenommen und die Transformation von B_{c_1} auf B_{c_2} betrachtet. Bei der Transformation $B_{y_{c_1}}$ auf $B_{x_{c_2}}$ im Induktionsschritt kann man schreiben:

$$\begin{pmatrix} \vec{f}_{*b'} \\ \vec{f}_{xb'} \end{pmatrix} \stackrel{!}{=} \Lambda_{y_{c_1}}^{x_{c_2}} \begin{pmatrix} \vec{f}_{*b} \\ \vec{f}_{yb} \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} \vec{f}_{*b} \\ \vec{f}_{yb} \end{pmatrix} \quad \text{mit } x, y \in \{0, 1\}.$$

Es gilt also:

$$\begin{aligned} \vec{f}_{*b'} &= A\vec{f}_{*b} + B\vec{f}_{yb} \\ \vec{f}_{xb'} &= C\vec{f}_{*b} + D\vec{f}_{yb} \end{aligned}$$

1. Fall $x = y$: Das erste Bit kann also vernachlässigt werden und damit gilt Folgendes:

$$\begin{aligned} \vec{f}_{*b'} &= \Lambda_{c_1}^{c_2} \vec{f}_{*b} \\ \vec{f}_{xb'} &= \Lambda_{c_1}^{c_2} \vec{f}_{xb}, \end{aligned} \tag{6.4}$$

also kann gesetzt werden:

$$\begin{aligned} A &= \Lambda_{c_1}^{c_2} & , & \quad B = 0 \\ C &= 0 & , & \quad D = \Lambda_{c_1}^{c_2}. \end{aligned}$$

Es ist

$$\Lambda_{x_{c_1}}^{x_{c_2}} = E \otimes \Lambda_{c_1}^{c_2}.$$

2. Fall $x \neq y$, also $y = \bar{x}$: Auch hier gilt die erste Formel von (6.4) und man kann setzen:

$$A = \Lambda_{c_1}^{c_2} & , \quad B = 0.$$

Dann gilt wegen $f_x = f_* - f_{\bar{x}}$:

$$\vec{f}_{xb} = \vec{f}_{*b} - \vec{f}_{\bar{x}b}$$

und es folgt

$$\begin{aligned} \vec{f}_{xb'} &= \Lambda_{c_1}^{c_2} (\vec{f}_{*b} - \vec{f}_{\bar{x}b}) \\ &= \Lambda_{c_1}^{c_2} \vec{f}_{*b} - \Lambda_{c_1}^{c_2} \vec{f}_{\bar{x}b}. \end{aligned}$$

Somit ist

$$C = \Lambda_{c_1}^{c_2}, \quad D = -\Lambda_{c_1}^{c_2},$$

und man erhält

$$\Lambda_{xc_1}^{\bar{x}c_2} = \Lambda_0^1 \otimes \Lambda_{c_1}^{c_2}.$$

Eine Abbildung ist durch die Bilder der Basiselemente eindeutig bestimmt. Deshalb ist $\Lambda_{yc_1}^{xc_2}$ die einzige Matrix, die die Transformation von B_{yc_1} auf B_{xc_2} beschreibt. \square

Transformation von Stringbasis auf die Walsh-Basis Nachdem nun die Transformationen von Stringbasis auf Bausteinsbasis und zurück sowie die Transformationen verschiedener Bausteinbasen elegant formuliert wurden, bleibt nun noch die Transformation von der Stringbasis auf die Walsh-Basis.

Voraussetzung für eine einfache Transformationsmatrix ist die Anordnung der Basisfunktionen und der Elemente des Suchraums bezüglich $11\dots 1$, also demjenigen Element, welches nur aus Einsen besteht. Eine Komponente der Transformationsmatrix Λ_W^l von der Stringbasis zur Walsh-Basis für die Zeichenkettenlänge l ist folgendermaßen bestimmt [Tou04]:

$$\lambda_{i,j} := (\Lambda_W^l)_{c_i, c_j} = (-1)^{|c_i \wedge c_j|} \quad c_i, c_j \in \mathfrak{A}.$$

Dabei ist $|c_i \wedge c_j|$ die Anzahl an Einsen, an denen c_i und c_j übereinstimmen. Für $l = 2$ ist Λ_W^2 zum Beispiel

$$\Lambda_W^2 = \begin{array}{c} 00 \\ 01 \\ 10 \\ 11 \end{array} \begin{array}{cccc} 00 & 01 & 10 & 11 \\ \left(\begin{array}{cccc} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{array} \right) = \left(\begin{array}{cc} 1 & 1 \\ 1 & -1 \end{array} \right)^{\otimes 2} = (\Lambda_W^1)^{\otimes 2}.$$

Dies führt zu folgenden Satz.

Satz 6.6 Die Transformationsmatrix Λ_W^l von der Stringbasis auf die Walsh-Basis zur Zeichenkettenlänge l erfüllt folgende Gleichung:

$$\Lambda_W^l = (\Lambda_W)^{\otimes l} \quad \text{mit} \quad \Lambda_W = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Für die Transformation in umgekehrter Richtung gilt:

$$(\Lambda_W^{\otimes l})^{-1} = \frac{1}{2^l} \Lambda_W^{\otimes l}.$$

6 Basen des Suchraums

Beweis: Es bietet sich ein Induktionsbeweis an. Der Anfang wurde bereits gezeigt. Es wird angenommen, dass

$$\Lambda_W^l = \Lambda_W^{\otimes l}.$$

Eine Komponente von Λ_W^{l+1} wird mit λ_{xc_i,yc_j} bezeichnet. Dabei können x und y die Werte 0 oder 1 annehmen. Die i bzw. j laufen von 1 bis 2^l und c_i und c_j bezeichnen die 2^l verschiedenen Elemente des Suchraums zur Zeichenkettenlänge l . Das Element xc_i gehört dann zu dem Element, das durch Anfügen von x an das i -te Element der Dimension l erzeugt wird. Für den Übergang von $l = 1$ zu $l = 2$ gilt also

$$\mathfrak{A} = \{0, 1\} \xrightarrow{l \rightarrow l+1} \mathfrak{A} = \underbrace{\{00, 01\}}_{0c_i} \underbrace{\{10, 11\}}_{1c_i} \quad i = 1, 2.$$

Somit können die Komponenten von Λ_W^{l+1} in vier Klassen aufgeteilt werden:

$$\lambda_{0c_i,0c_j}, \quad \lambda_{0c_i,1c_j}, \quad \lambda_{1c_i,0c_j}, \quad \lambda_{1c_i,1c_j} \quad i, j = 1, \dots, 2^l.$$

Diese Notation erleichtert die Formulierung, nun kann man schreiben:

$$\lambda_{0c_i,0c_j} = \lambda_{c_i,c_j},$$

da sich die Anzahl der übereinstimmenden Einsen beim Übergang von l zu $l + 1$ nicht ändert. Mit demselben Argument folgt:

$$\lambda_{0c_i,1c_j} = \lambda_{c_i,c_j} \quad \text{und} \quad \lambda_{1c_i,0c_j} = \lambda_{c_i,c_j}.$$

Für $\lambda_{1c_i,1c_j}$ gilt:

$$\lambda_{1c_i,1c_j} = -\lambda_{c_i,c_j},$$

da sich die Anzahl der übereinstimmenden 1-Bits um Eins erhöht:

$$\lambda_{1c_i,1c_j} = (-1)^{|1c_i \wedge 1c_j|} = (-1)^{1+|c_i \wedge c_j|} = -\lambda_{c_i,c_j}.$$

Für die Transformationsmatrix folgt dann:

$$\Lambda_W^{l+1} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \otimes \Lambda_W^l = \Lambda_W^{\otimes(l+1)}.$$

Die Inverse zu Λ_W^{l+1} ist

$$\begin{aligned} (\Lambda_W^{\otimes(l+1)})^{-1} &= (\Lambda_W^{-1})^{\otimes(l+1)} \\ &= \frac{1}{2^{l+1}} \Lambda_W^{\otimes(l+1)}, \text{ da} \\ \Lambda_W^{-1} &= \frac{1}{2} \Lambda_W. \end{aligned}$$

6.4 Die Basen des Dualraums $Fun^*(\mathfrak{A})$

Mit den in Abschnitt 6.3 hergeleiteten Übergangsmatrizen ist es nun sehr einfach, die Dualbasen zu berechnen, denn sind B_i und B_a zwei Basen und B^i und B^a die dazu dualen Basen, dann gilt mit der Übergangsmatrix Λ_i^a von Basis B_a auf B_i für die Basiselemente $e_i \in B_i$ und $e_a \in B_a$

$$e_i = \Lambda_i^a e_a$$

auch

$$e^i = e^a (\Lambda_i^a)^{-1},$$

wenn $e^i \in B^i$ und $e^a \in B^a$.

Vertex-Basis Betrachtet wird die duale Basis von B_S . Auf Grund der Einbettung von \mathfrak{A} in den Dualraum $Fun^*(\mathfrak{A})$ wird sie in [CS] die Vertex-Basis genannt, da die duale Basisfunktion $g_i \in Fun^*(\mathfrak{A})$ den Funktionswert von $f \in Fun(\mathfrak{A})$ an dem i -ten Element wiedergibt (siehe Abschnitt 6.1).

Die Basis wird mit B_V bezeichnet und ihre Elemente mit g_{x_i} , wobei x_i ein Element aus \mathfrak{A} ist:

$$g_{x_i}(f) = f(x_i) \quad \forall x_i \in \mathfrak{A}, i = 1, \dots, N.$$

Taylor-Basis In [CS] wird gezeigt, dass die Taylor-Basis B_T die duale Basis zur Bausteinbasis ist. Es bezeichne g_T den Vektor der Basisfunktionen von B_T und g_V den der Basisfunktionen von B_V . Es gilt:

$$\begin{aligned} g_T &= g_V (\Lambda^{\otimes l})^{-1} \\ &= g_V (\Lambda^{-1})^{\otimes l}. \end{aligned}$$

Walsh-Basis Die duale Basis zur Walsh-Basis bringt wenig Neues, denn

$$(\Lambda_W^{\otimes l})^{-1} = \frac{1}{2^l} \Lambda_W^{\otimes l}.$$

6.5 Transformation der Wahrscheinlichkeitsvektoren

Die Evolutionsgleichungen werden mit Hilfe der Wahrscheinlichkeiten von Elementen des Suchraums ausgedrückt.

Es besteht ein Isomorphismus zwischen der Menge der charakteristischen Funktionen über \mathfrak{A} , im Folgenden $CF(\mathfrak{A})$ genannt, und dem Wahrscheinlichkeitsraum über \mathfrak{A} . Wenn man sich also auf diejenigen Teilmengen von \mathfrak{A} beschränkt, die als Träger der Basisfunktionen dienen, so transformieren sich die Wahrscheinlichkeitsvektoren beim Basiswechsel auf die gleiche Art wie die Vektoren der Basisfunktionen. Denn die Basisfunktionen sind ausschließlich charakteristische Funktionen.

6 Basen des Suchraums

Betrachtet wird die Isomorphie zwischen $CF(\mathfrak{A})$ und dem Wahrscheinlichkeitsraum $(\mathfrak{A}, \mathfrak{P}(\mathfrak{A}), P)$. Der Raum der charakteristischen Funktionen wird mit der Addition \oplus ausgestattet:

$$f_A \oplus f_B := f_{A \cup B} = f_A + f_B - f_{A \cap B} \quad \forall A, B \subset \mathfrak{A}.$$

Der Wahrscheinlichkeitsraum erhält folgende Addition \odot :

$$P(A) \odot P(B) := P(A \cup B) = P(A) + P(B) - P(A \cap B) \quad \forall A, B \subset \mathfrak{A}.$$

Dann gibt es eine Isomorphie γ :

$$\begin{aligned} \gamma : (CF(\mathfrak{A}), \oplus) &\longrightarrow ((\mathfrak{A}, \mathfrak{P}(\mathfrak{A}), P), \odot) \\ f_A &\mapsto P(A) \end{aligned}$$

Bei γ handelt es sich um einen Homomorphismus, denn

$$\begin{aligned} \gamma(f_A \oplus f_B) &= \gamma(f_{A \cup B}) \\ &= P(A \cup B) \\ &= P(A) \odot P(B). \end{aligned}$$

Der Homomorphismus γ ist bijektiv, da jede Teilmenge von \mathfrak{A} genau eine charakteristische Funktion und genau eine Wahrscheinlichkeit definiert.

Betrachtet man nun statt der charakteristischen Funktionen, die als Basisfunktionen dienen, die Wahrscheinlichkeiten, die mit ihren Trägern assoziiert sind, dann transformieren sich die Vektoren der Wahrscheinlichkeiten genauso wie die Basisfunktionen. Für den Vektor P , dessen i -te Komponente die Wahrscheinlichkeit des Auftretens des i -ten Elementes von \mathfrak{A} in der Population angibt, gilt also:

$$\begin{aligned} P_B &= \Lambda^{\otimes l} P \\ P_W &= \Lambda_W^{\otimes l} P. \end{aligned}$$

Der Vektor P_B bzw. P_W ist der Populationsvektor bezüglich der Bausteinbasis bzw. der Walsh-Basis. So bedeutet die i -te Komponente von P_B die Wahrscheinlichkeit, dass der i -te Baustein in der Population vertreten ist. Diese ergibt sich aus der Summe der Wahrscheinlichkeiten der Elemente, die im i -ten Baustein liegen.

6.6 Tensorschreibweise der genetischen Operatoren

Die Wirkung der genetischen Operatoren kann auch mit Hilfe von Matrizen beschrieben werden. Damit werden die Basiswechsel erleichtert. Es interessiert, wie sich diese Matrizen bei einem Basiswechsel verhalten.

Selektion Begonnen wird mit der Selektion. In Matrizenschreibweise kann sie folgendermaßen beschrieben werden [CS]:

$$P' = FP.$$

Dabei ist P der N -dimensionale Populationsvektor, der die relativen Häufigkeiten der einzelnen Elemente in der Population beinhaltet. Das heißt, die i -te Komponente von P gibt die relative Häufigkeit des i -ten Elementes von \mathfrak{A} an:

$$P = \begin{pmatrix} P(x_1) \\ \vdots \\ P(x_N) \end{pmatrix} \quad \sum_{x_i \in \mathfrak{A}} P(x_i) = \sum_{x_i \in P_t} P(x_i) = 1,$$

wobei jetzt der Einfachheit halber auf den Zeitindex t verzichtet wird. Der Vektor P' besteht aus den relativen Häufigkeiten der Elemente nach der Selektion.

Die Matrix F ist eine diagonale Fitnessmatrix. Die Komponente f_{ii} ist der relative Fitnesswert des i -ten Elementes multipliziert mit der Populationsgröße.

In Abschnitt 6.5 wurde gezeigt, dass sich die Wahrscheinlichkeitsvektoren wie die Basenvektoren transformieren. Bei einem Basiswechsel auf eine andere Basis B_A durch die Übergangsmatrix Λ_A folgt für die Fitnessmatrix F_A bezüglich der Basis B_A aus:

$$P'_A = \Lambda_A P' = \Lambda_A (FP) \quad \text{und} \quad P'_A = F_A P_A = F_A (\Lambda_A P),$$

dass

$$F_A = \Lambda_A F \Lambda_A^{-1}. \tag{6.5}$$

Auf Grund dieses Transformationsverhaltens ist F ein Tensor vom Typ (1,1) [CS].

Mutation Als Nächstes wird die Mutation betrachtet. Für $l = 1$ bezüglich der Stringbasis ist dann [CS]

$$P_m = M_l P$$

mit P_m als Vektor mit den relativen Häufigkeiten der Elemente nach der Mutation. Für $l = 1$ ist

$$M_1 = \begin{pmatrix} 1 - p_m & p_m \\ p_m & 1 - p_m \end{pmatrix},$$

wenn p_m die Mutationswahrscheinlichkeit eines Bits ist. Die Bedeutung der Komponente $(M_l)_{ij}$ ist Folgende. Mit $(M_l)_{ij}$ wird die Wahrscheinlichkeit angegeben, mit der das j -te Element in das i -te Element von \mathfrak{A} mutiert wird. Bei einer Stringlänge von l gilt also

$$(M_l)_{ij} = p_m^{d_{ij}^H} (1 - p_m)^{l - d_{ij}^H}$$

6 Basen des Suchraums

mit d_{ij}^H als Hamming-Distanz zwischen dem i -ten und dem j -ten Element. Für $l = 2$ und nach Konvention geordnetes \mathfrak{A} ist M_2 also

$$\begin{aligned} M_2 &= \begin{matrix} & & 11 & 10 & 01 & 00 \\ \begin{matrix} 11 \\ 10 \\ 01 \\ 00 \end{matrix} & \begin{pmatrix} (1-p_m)^2 & p_m(1-p_m) & p_m(1-p_m) & p_m^2 \\ p_m(1-p_m) & (1-p_m)^2 & p_m^2 & p_m(1-p_m) \\ p_m(1-p_m) & p_m^2 & (1-p_m)^2 & p_m(1-p_m) \\ p_m^2 & p_m^2 & p_m(1-p_m) & (1-p_m)^2 \end{pmatrix} \end{matrix} \\ &= M_1 \otimes M_1, \end{aligned}$$

wobei zur Erleichterung die entsprechenden Elemente von \mathfrak{A} klein an Zeilen und Spalten notiert wurden. Der Suchraum wurde bezüglich 00 geordnet. Die Gleichung führt zu folgenden Satz.

Hilfssatz 6.7 *Für die Matrix, die die Wirkung der Mutation bezüglich der Stringbasis beschreibt, gilt [CS]*

$$M_l = (M_1)^{\otimes l}. \quad (6.6)$$

Beweis: Zunächst wird die Formel bezüglich der Stringbasis bewiesen. Es bietet sich ein Induktionsbeweis an. Der Anfang wurde oben bereits getan, Induktionsvoraussetzung ist (6.6). Betrachtet wird nun der Induktionsschritt von $l \rightarrow l+1$:

$$\begin{aligned} P_m &= M_{l+1}P \\ \Rightarrow \begin{pmatrix} \vec{P}_{m\bar{x}c} \\ \vec{P}_{mxc} \end{pmatrix} &= \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} \vec{P}_{\bar{x}c} \\ \vec{P}_{xc} \end{pmatrix}. \end{aligned}$$

Die Matrizen sind vom Typ $N \times N$, die Vektoren bestehen nach Notationsvereinbarung aus:

$$\begin{aligned} \vec{P}_{\bar{x}c} &= \left(P(\bar{x}c_1), \dots, P(\bar{x}c_N) \right)^T & c_i \in \mathfrak{A} \\ \vec{P}_{xc} &= \left(P(xc_1), \dots, P(xc_N) \right)^T \\ \vec{P}_{m\bar{x}c} &= \left(P_m(\bar{x}c_1), \dots, P_m(\bar{x}c_N) \right)^T \\ \vec{P}_{mxc} &= \left(P(xc_1), \dots, P(xc_N) \right)^T. \end{aligned}$$

Man erhält also folgende Gleichungen:

$$\begin{aligned} \vec{P}_{m\bar{x}c} &= A\vec{P}_{\bar{x}c} + B\vec{P}_{xc} \\ \vec{P}_{mxc} &= C\vec{P}_{\bar{x}c} + D\vec{P}_{xc}. \end{aligned}$$

Ein Element $\bar{x}c_j$ mutiert in ein Element $\bar{x}c_i$ mit einer Wahrscheinlichkeit von $(1-p_m)(M_l)_{ij}$, da das erste Bit gleich ist und c_j in c_i mit der Wahrscheinlichkeit $(M_l)_{ij}$ mutiert wird. Das Element $\bar{x}c_j$ mutiert in xc_j mit einer

6.6 Tensorschreibweise der genetischen Operatoren

Wahrscheinlichkeit von $p_m(M_l)_{ij}$, da sie sich im ersten Bit unterscheiden. Also gilt:

$$\begin{aligned} \begin{pmatrix} \vec{P}_{m\bar{x}c} \\ \vec{P}_{mxc} \end{pmatrix} &= \begin{pmatrix} (1-p_m)M_l & p_m M_l \\ p_m M_l & (1-p_m)M_l \end{pmatrix} \begin{pmatrix} \vec{P}_{\bar{x}c} \\ \vec{P}_{xc} \end{pmatrix} \\ \Rightarrow P_m &= (M_1)^{\otimes l+1} P. \end{aligned}$$

□

Bei einem Basiswechsel auf die Basis B_A folgt analog der Argumentation für die Selektion:

$$(M_l)_A = \Lambda_A M_l \Lambda_A^{-1}. \quad (6.7)$$

Also ist damit auch M_l ein Tensor vom Typ $(1,1)$.

Die Matrix, die die Wirkung der Mutation beschreibt, erfüllt (6.6) unabhängig davon, bezüglich welcher Basis M_l gegeben ist. Für diese Basisunabhängigkeit wird gezeigt, dass die Tensormultiplikation mit der Matrizenmultiplikation verträglich ist:

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD).$$

Diese Gesetzmäßigkeit ist für 2×2 -Matrizen übersichtlicher zu zeigen. Sie überträgt sich aber ohne Umstände auf $N \times N$ -Matrizen:

$$\begin{aligned} (A \otimes B)(C \otimes D) &= \begin{pmatrix} a_{11}B & a_{12}B \\ a_{21}B & a_{22}B \end{pmatrix} \begin{pmatrix} c_{11}D & c_{12}D \\ c_{21}D & c_{22}D \end{pmatrix} \\ &= \begin{pmatrix} (a_{11}c_{11} + a_{12}c_{21})BD & (a_{11}c_{12} + a_{12}c_{22})BD \\ (a_{21}c_{11} + a_{22}c_{21})BD & (a_{21}c_{12} + a_{22}c_{22})BD \end{pmatrix} \\ &= \begin{pmatrix} (AC)_{11}BD & (AC)_{12}BD \\ (AC)_{21}BD & (AC)_{22}BD \end{pmatrix} \\ &= (AC) \otimes (BD). \end{aligned}$$

Bezeichnet Λ_A die Übergangsmatrix von der Stringbasis zu einer anderen Basis B_A und $(M_l)_A$ die Mutationsmatrix bezüglich B_A , so gilt wegen oben also:

$$\begin{aligned} (M_l)_A &= (\Lambda_A)^{\otimes l} M_1^{\otimes l} (\Lambda_A^{-1})^{\otimes l} \\ &= (\Lambda_A M_1 \Lambda_A^{-1})^{\otimes l}. \end{aligned}$$

Damit ist (6.6) also für alle Basen gültig.

Rekombination Nun folgt die Rekombination. Der Ausdruck, der die Wirkung der Rekombination in der Stringbasis beschreibt, ist

$$P_c(I) = (1 - p_c)P'(I) + p_c \sum_M \sum_{J,K} p(M) \lambda_I^{JK}(M) P'(J) P'(K) \quad (6.8)$$

und gibt die Wahrscheinlichkeit an, mit der die Zeichenkette I nach der Rekombination vorhanden ist. Der erste Term von (6.8) beschreibt die

6 Basen des Suchraums

Wahrscheinlichkeit, mit der I nicht gekreuzt wird, der Ausdruck $P'(I)$ ist der Anteil von I in der Population nach der Selektion. Der zweite Term gibt die Wahrscheinlichkeit an, mit der I nach dem tatsächlichen Stattfinden der Rekombination von je zwei Eltern J und K vorhanden ist. Die erste Summe läuft über alle möglichen Rekombinationsmasken M , die zweite Summe über alle Elternkombinationen.

Der Faktor $\lambda_I^{JK}(M)$ ist genau dann Eins, wenn aus der Kreuzung von J und K bei Verwendung der Maske M die Zeichenkette I als erster Nachkomme hervorgeht. Dabei bedeutet eine 0 an der i -ten Stelle der Maske die Übernahme des i -ten Allels des ersten Partners, eine 1 die Vererbung durch den zweiten Partner. Für zwei komplementäre Masken $M = m_1 m_2 \dots m_l$ und $\bar{M} = \bar{m}_1 \bar{m}_2 \dots \bar{m}_l$ mit $\bar{m}_i = 1 - m_i$ gilt dann also

$$\lambda_I^{JK}(M) = \lambda_I^{KJ}(\bar{M}),$$

da sich die Rollen der Eltern vertauschen.

Beim Basiswechsel auf eine Basis B_A ist nur der zweite Term von (6.8) interessant. Für ein Element I' bezüglich der Basis B_A und fester Maske M hat man

$$\begin{aligned} P_A(I') &= \sum_{I,J,K} (\Lambda_A)_{I'}^I \left(\lambda_I^{JK} P'(J) P'(K) \right) \\ &\stackrel{!}{=} \sum_{J',K'} (\lambda_A)_{I'}^{J'K'} P'_A(J') P'_A(K') \\ &= \sum_{J',K'} (\lambda_A)_{I'}^{J'K'} \sum_{J,K} (\Lambda_A)_{J'}^J (\Lambda_A)_{K'}^K P'(J) P'(K) \\ \Rightarrow (\lambda_A)_{I'}^{J'K'} &= (\Lambda_A)_{I'}^I \lambda_I^{JK} (\Lambda_A^{-1})_{J'}^J (\Lambda_A^{-1})_{K'}^K. \end{aligned}$$

Somit ist λ ein Tensor der Stufe (2,1) [CS]. Dabei wurde der Übersichtlichkeit halber das Argument M von λ und λ_A weggelassen.

6.7 Die Wirkung der genetischen Operatoren bezüglich der verschiedenen Basen

Im Folgenden werden die genetischen Operatoren bezüglich der eingeführten Basen untersucht. Die Basen beschreiben die Wirkung der genetischen Operatoren verschieden gut. So eignet sich zum Beispiel die Stringbasis für die Selektion am besten, die Bausteinbasis für die Rekombination und die Walsh-Basis für die Mutation [CS].

Selektion Betrachtet man die Selektion in der Bausteinbasis, das heißt

$$P'_B = F_B P_B,$$

so weiß man wegen (6.5), dass

$$F_B = \Lambda F \Lambda^{-1}.$$

6.7 Die Wirkung der genetischen Operatoren bezüglich der verschiedenen Basen

Schon im eindimensionalen Fall ist F_B nicht mehr diagonal:

$$\begin{aligned} F_B &= \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} f_1 & 0 \\ 0 & f_2 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} f_1 & -f_1 \\ 0 & f_2 \end{pmatrix} \\ &= \begin{pmatrix} f_1 & f_2 - f_1 \\ 0 & f_2 \end{pmatrix}. \end{aligned}$$

Dies ist leicht zu erklären. In der Stringbasis steht ein Basiselement für genau ein Individuum, während in der Bausteinbasis ein Basiselement mehrere Individuen vereinen kann. Daher kann F_B nicht diagonal sein.

Laut [CS] kann die Matrix F_B immer in $F_B = F'_B + A$ zerlegt werden, wobei $AP_B = 0$ gilt und F'_B eine Diagonalmatrix ist. Damit kann die Selektion auch bezüglich einer Bausteinbasis nach der Zerlegung in F'_B und A durch eine Diagonalmatrix beschrieben werden.

Die Matrix F_W , die die Selektion in der Walsh-Basis beschreibt ist sehr kompliziert, so hängt zum Beispiel die Anzahl der von 0 verschiedenen Elemente von dem Grad der Epistasie der Landschaft ab [CS]. Die Epistasie ist die Interaktion zweier Gene bei der Ausprägung des Phänotyps. Mathematisch beschreibt die Epistasie damit die Korrelationen zwischen den verschiedenen Bits.

Für die Selektion eignet sich also die Beschreibung in der Stringbasis am besten, da die Matrix, die die Wirkung der Selektion beschreibt, dann die einfachste Gestalt besitzt.

Mutation Für die die Baustein- und Walsh-Basis gelten wegen (6.7):

$$\begin{aligned} M_B &= \Lambda M_l \Lambda^{-1} \\ &= \begin{pmatrix} 1 & 0 \\ p & 1 - 2p_m \end{pmatrix} \end{aligned}$$

und

$$\begin{aligned} M_W &= \Lambda_W M_l \Lambda_W \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 - 2p_m \end{pmatrix}. \end{aligned}$$

Die Einfachheit von M_W überträgt sich auch ins l -dimensionale, denn es gilt für den l -dimensional Fall nach dem Hilfsatz 6.7

$$M_l = M^{\otimes l}$$

unabhängig von der Basis.

Auch die Bausteinbasis eignet sich mit dieser einfachen Matrixstruktur für die Beschreibung des Mutationseffektes, die Walsh-Basis mit der noch einfacheren Diagonalstruktur jedoch noch mehr.

6 Basen des Suchraums

Rekombination Bei der Rekombination wird der (2,1)-Tensor λ betrachtet. Für den zweidimensionalen Fall sind zum Beispiel:

$$\begin{array}{l} \lambda_{11}(00) = \lambda_{11}^T(11) = \\ \lambda_{11}(01) = \lambda_{11}^T(10) = \end{array} \begin{array}{c} \begin{array}{c} \text{1.Elter:} \\ \text{00} \\ \text{01} \\ \text{10} \\ \text{00} \end{array} \end{array} \begin{array}{c} \begin{array}{c} \text{2.Elter:} \\ \text{00} \\ \text{01} \\ \text{10} \\ \text{11} \end{array} \end{array} \begin{array}{c} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix} \end{array} \quad \text{und}$$

Die Elemente des Suchraumes korrespondieren zu bestimmten Zeilen und Spalten und werden daher wieder klein an die Matrix notiert. Dabei korrespondiert das erste Elter zu den Zeilen, das zweite Elter zu den Spalten. Das Matrixelement $(\lambda_{11}(00))_{ij}$ ist demnach genau dann 1, wenn sich aus dem ersten Elter x_i und dem zweiten Elter x_j mit der Maske 00 das Element 11 rekombinieren lässt.

Die Reihenfolge der Elemente richtet sich wieder nach der vereinbarten Anordnung. Dabei ist das Bezugselement in diesem Beispiel 11.

Zur Vereinfachung kann das Mittel über alle Masken gebildet werden:

$$\lambda = \sum_M p(M) \lambda(M),$$

dabei ist $p(M)$ die Wahrscheinlichkeit, mit der Maske M verwendet wird. Man erhält, wenn alle Masken gleichwahrscheinlich sind:

$$\begin{array}{l} \lambda_{00} = \frac{1}{4} \begin{pmatrix} 4 & 2 & 2 & 1 \\ 2 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \\ \lambda_{01} = \frac{1}{4} \begin{pmatrix} 0 & 2 & 0 & 1 \\ 2 & 4 & 1 & 2 \\ 0 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 \end{pmatrix} \end{array} \quad \begin{array}{l} \lambda_{11} = \frac{1}{4} \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 2 \\ 1 & 2 & 2 & 4 \end{pmatrix} \\ \lambda_{10} = \frac{1}{4} \begin{pmatrix} 0 & 0 & 2 & 1 \\ 0 & 0 & 1 & 0 \\ 2 & 1 & 4 & 2 \\ 1 & 0 & 2 & 0 \end{pmatrix}. \end{array}$$

Bezüglich der Bausteinbasis sieht λ_I^B folgendermaßen aus:

$$\begin{array}{l} \lambda_{00}^B = \frac{1}{4} \begin{pmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\ \lambda_{01}^B = \frac{1}{4} \begin{pmatrix} 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \quad \begin{array}{l} \lambda_{11}^B = \frac{1}{4} \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \\ \lambda_{10}^B = \frac{1}{4} \begin{pmatrix} 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \end{array}$$

6.8 Zusammenhang zwischen Bausteinbasis und Renormierungsgruppe

Allgemein kann man die Rekombination mit einem Tensorprodukt für beliebige Dimension l beschreiben. Für eine beliebige Zeichenkette $I = x_1 \dots x_l$ gilt bezüglich der Stringbasis [CS]:

$$\lambda_I = \lambda_{x_1 x_2 \dots x_l} = \lambda_{x_1} \otimes \lambda_{x_2} \otimes \dots \otimes \lambda_{x_l} \quad \text{mit}$$

$$\lambda_0 = \frac{1}{2} \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix} \quad \lambda_1 = \frac{1}{2} \begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix},$$

bzw. in der Bausteinbasis:

$$\lambda_{b_1 b_2 \dots b_l}^B = \lambda_{b_1}^B \otimes \lambda_{b_2}^B \otimes \dots \otimes \lambda_{b_l}^B \quad \text{mit}$$

$$\lambda_*^B = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \lambda_1^B = \frac{1}{2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Man erkennt, dass die Struktur von λ_I^B einfacher ist als λ_I . Daher eignet sich zur Beschreibung der Rekombination die Bausteinbasis mehr als die Stringbasis.

Die Darstellung der Rekombination in der Walsh-Basis ist kompliziert. Diese findet man in [Wri99].

6.8 Zusammenhang zwischen Bausteinbasis und Renormierungsgruppe

Nachdem nun mehrere Basen betrachtet wurden und sie bezüglich ihrer Eignung für die Behandlung der Evolution im Genetischen Algorithmus untersucht wurden, wird sich nun der Bausteinbasis gewidmet. Sie taucht im Zusammenhang mit dem „Coarse Graining“ auf, welches eine Methode zur Behandlung von dynamischen Systemen ist, die häufig in der Statistischen Mechanik verwendet wird.

Die Bezeichnung „Coarse Graining“ ist ein Sammelbegriff für Vereinfachungsprozeduren. Eine mögliche Übersetzung wäre „Vergrößerung“. So versucht man die Auflösung, mit der man ein System betrachtet, zu vergrößern und somit durch Ausblenden irrelevanter Details die Untersuchungen daran zu vereinfachen und sich auf wesentliche Eigenschaften zu konzentrieren.

In der Statistischen Mechanik findet man eine solche Vergrößerungsmethode in der Renormierungsgruppe. Diese Namensgebung ist jedoch irreführend, da es sich zum einen um keine eindeutig bestimmte Gruppe handelt, sondern im Gegenteil meist mehrere mögliche für sie gibt. Und zum anderen liegt der Renormierungsgruppe keine Gruppeneigenschaft zu Grunde, sondern nur die Halbgruppeneigenschaft.

Die Renormierungsgruppe (RG) ist ein System von Koordinatentransformationen, das die Existenz von Fixpunkten einer Dynamik klären kann. Die RG-Transformationen werden dabei so gewählt, dass die funktionelle Form der transformierten Gleichung erhalten bleibt (Forminvarianz), denn

6 Basen des Suchraums

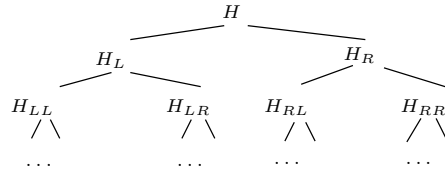


Abbildung 6.4: Struktur der Vergrößerung für die Evolutionsgleichung

anschließend wird die RG-Transformation iteriert, wodurch man einen Fixpunkt der Dynamik entdecken kann. Nämlich an der fehlenden Auswirkung der Iteration auf die Gleichung. Durch die Iteration erhält man eine Folge von Gleichungen mit immer höherer Vergrößerung.

Die Transformationen auf eine Bausteinbasis bilden eine Renormierungsgruppe. Mit ihrer Hilfe findet nicht nur eine Reduktion der Freiheitsgrade statt. Die Transformationen auf Schemen bzw. Bausteine ist überdies die einzige Vereinfachung, die die Evolutionsgleichung forminvariant lässt [SS02], [Vos99].

Eine Vermutung von Stephens ist, dass die Bausteine die wesentlichen Freiheitsgrade („effective degrees of freedom“) eines Genetischen Algorithmus sind. Also jene Freiheitsgrade, auf die ein System minimal reduziert werden kann [SH98].

Versucht man die Evolutionsgleichung (3.17) zu lösen, so erhält man ein System von Gleichungen mit immer höherem Grad an Vergrößerung (siehe Abbildung 6.4), denn zur Berechnung von $P(H, t + 1)$ benötigt man die entsprechenden Terme von $P'(H_L, t)$ und $P'(H_R, t)$.

Diese hierarchische Ordnung zusammen mit der Forminvarianz und der Reduktion der Freiheitsgrade begründen nicht allein die Eigenschaft als Renormierungsgruppe. Die Menge der Transformationen von der Stringbasis auf eine Bausteinbasis sowie von einer Bausteinbasis auf eine andere bildet eine Gruppe.

Folgende Notation wird verwendet: $P(x_1, x_2)$ bezeichne den Vektor der Basenelemente bezüglich der Basis B_{x_1} , wobei die Anordnung der Komponenten bezüglich x_2 erfolgt. So ist zum Beispiel $P(S, x_1)$ der Vektor der Stringbasiselemente, die bezüglich x_1 angeordnet sind. Es bezeichne Λ_x die Übergangsmatrix von der Stringbasis auf die Bausteinbasis B_x , wobei die Basenelemente bezüglich x angeordnet sind. Es ist aus Abschnitt 6.3 bekannt:

$$\begin{aligned}\Lambda_{x_1} P(S, x_1) &= P(x_1, x_1) \\ \Lambda_{x_2} P(S, x_2) &= P(x_2, x_2)\end{aligned}$$

Es gibt eine Übergangsmatrix $\Lambda_{x_1}^{x_2}$ mit

$$\Lambda_{x_1}^{x_2} P(x_1, x_1) = P(x_2, x_2)$$

und es gilt

$$\Lambda_{x_2} = \Lambda_{x_1}^{x_2} \Lambda_{x_1}.$$

6.8 Zusammenhang zwischen Bausteinbasis und Renormierungsgruppe

Die binäre Operation ist also gegeben. Das hintereinander ausführen von Basiswechseln ist assoziativ. Zu jeder Transformation gibt es eine Rücktransformation. Das neutrale Element ist die Transformation von einer Basis auf sich selbst, wird also durch die Einheitsmatrix beschrieben. Demnach ist die Menge der Transformationen von der Stringbasis auf eine Bausteinbasis sowie von einer Bausteinbasis auf eine andere eine Gruppe.

Welchen Nutzen kann man nun aus der Erkenntnis ziehen, dass die Transformationen auf und zwischen Bausteinbasen eine Renormierungsgruppe bilden? Fixpunkte im dynamischen System sollten erkennbar sein.

Es wird ein Beispiel für drei Dimensionen betrachtet. Die Mutation wird der Einfachheit halber weggelassen. Die Rekombinationsmasken sind alle gleichwahrscheinlich. Die Evolutionsgleichung für den Baustein 111 lautet dann:

$$\begin{aligned}
 P_{t+1}(111) &= (1 - p_c)P'_t(111) + \frac{p_c}{4} \sum_M P'_t(H_L)P'_t(H_R) \\
 &= (1 - p_c)P'_t(111) + \frac{p_c}{4} P'_t(11*)P'_t(**1) + \frac{p_c}{4} P'_t(1*1)P'_t(*1*) \\
 &\quad + \frac{p_c}{4} P'_t(1***)P'_t(*11) + \frac{p_c}{4} P'_t(111)P'_t(***)
 \end{aligned} \tag{6.9}$$

Der Term $P'_t(11*)$, der für (6.9) benötigt wird, wird näher betrachtet:

$$\begin{aligned}
 P'_t(11*) &= 2nf_t(11*)P_{t-1}(11*) \\
 &= 2nf_t(11*) \left((1 - p_c)P'_t(11*) + \frac{p_c}{2} P'_{t-1}(11*)P'_{t-1}(***) \right. \\
 &\quad \left. + \frac{p_c}{2} P'_{t-1}(1***)P'_{t-1}(*1*) \right) \\
 &= 2nf_t(11*) \left(\left(1 - \frac{p_c}{2}\right) P'_t(11*) + \frac{p_c}{2} P'_{t-1}(1***)P'_{t-1}(*1*) \right)
 \end{aligned} \tag{6.10}$$

Die Gleichung für $P'_{t-1}(1**)$ ist:

$$\begin{aligned}
 P'_{t-1}(1**) &= 2nf_{t-1}(1**)P_{t-2}(1**) \\
 &= 2nf_{t-1}(1**) \left((1 - p_c)P'_{t-2}(1**) \right. \\
 &\quad \left. + p_c P'_{t-2}(1**)P'_{t-2}(***) \right) \\
 &= 2nf_{t-1}(1**)
 \end{aligned}$$

Die Vergrößerung endet also bei den 1-Schemen, denn die Rekombination hat keinerlei Auswirkungen auf die Verteilung der 1-Schemen [SW99]. Das Gleichgewicht für (6.9) kann demnach nur dann erreicht werden, wenn die Gleichgewichtsbedingung für die 1-Schemen erfüllt ist. Denn gilt für die 1-Schemen $1**$ und $*1*$:

$$\begin{aligned} f_t(1**) &= f_{t-1}(1**) && \text{bzw.} \\ f_t(*1*) &= f_{t-1}(*1*) && \forall t > t_0, \end{aligned}$$

dann ändert sich auch $f_t(11^*)$ nicht mehr, da sich der Anteil der 1-Schemen in der Population nicht mehr ändert und demnach die relative Fitness von 11^* stagniert. Bleibt $f_{t-1}(11^*)$ unverändert, so stagniert auch $P'_t(11^*)$ und (6.10) kann seinen Fixpunkt erreichen. Ist die Gleichgewichtsverteilung nicht nur für 11^* sondern für alle 2-Schemen erreicht, so ändert sich auch $P_{t+1}(111)$ nicht mehr. Der Gleichgewichtszustand der Population wird also erst erreicht, wenn er für die 1-Schemen erreicht wurde [SW99].

Ein weiterer Grund für die Bedeutung der Bausteinbasen ist folgender: In [Ste03] werden für die Evolutionsgleichung bezüglich einer Bausteinbasis Feynman-Regeln für ein Feynman-Diagramm aufgestellt. Ein Feynman-Diagramm ermöglicht die Berechnung aller Komponenten der S-Matrix. In der Physik beschreibt diese Matrix die Abbildung aller möglichen Anfangszustände in die Menge aller möglichen Endzustände, die durch eine gegebene Wechselwirkung erzeugt werden. In unserem Fall wird die Wechselwirkung durch die genetischen Operatoren definiert.

6.9 Zusammenfassung

Dieses Kapitel hat sich mit der mathematischen Beschreibung des Suchraumes beschäftigt und es konnten Schlussfolgerungen aus der Formalisierung für die Behandlung der Dynamik in Genetischen Algorithmen gezogen werden. Es basierte hauptsächlich auf den Veröffentlichungen [SH98], [SW99] und [CS].

Verschiedene Basen wie die Stringbasis, die Bausteinbasen und die Walsh-Basis wurden eingeführt und untersucht. Geometrische sowie analytische Transformationen wurden hergeleitet. Weiterhin wurde untersucht, welche Basis für welchen genetischen Operator die angebrachteste ist. Dabei wird die Selektion am natürlichsten in der Stringbasis beschrieben, die Rekombination in einer Bausteinbasis und die Mutation in der Walsh-Basis.

Zu guter Letzt konnte eine Analogie zur Statistischen Mechanik aufgezeigt werden, denn die Transformationen auf und zwischen den Bausteinbasen ist als Renormierungsgruppe zur mathematischen Behandlung der Evolution im Genetischen Algorithmus anwendbar. Mit ihrer Hilfe kann ein Fixpunkt der Evolutionsgleichung gefunden werden. Weiterhin ermöglichen die Bausteinbasen das Aufstellen von Feynman-Regeln für Feynman-Diagramme.

7 Simulationsergebnisse

In diesem Kapitel werden numerische Ergebnisse vorgestellt. Es interessiert, inwieweit sich Vermutungen bestätigen lassen und welche Auswirkungen bestimmte Änderungen, wie zum Beispiel der gerichtete Rekombinationsoperator, tatsächlich haben.

In den Simulationen wird ein Genetischer Algorithmus verwendet, der die Symmetrie als Fitnessfunktion benutzt. Es wird untersucht, welche Auswirkungen bestimmte Parameter auf die Durchschnittssymmetrie haben. Mit der Durchschnittssymmetrie ist der Durchschnitt über die prozentuale Symmetrie aller Zeichenketten in der Endpopulation gemeint:

$$\text{Durchschnittssymmetrie} = \frac{1}{2n} \sum_{I \in P_t} \frac{F(I)}{l}.$$

Die Populationsgröße ist $2n$. Die Durchschnittssymmetrie einer zufälligen Population ist 25%, wie bereits mit (4.1) festgestellt wurde. Die Symmetrie bewegt sich also zwischen den Werten 0.25 und 0.5. In den Graphen, die in diesem Kapitel gezeigt werden, wird die Durchschnittssymmetrie über die letzten 500 Generationen gemittelt, um unerwünschte Schwankungen zu vermeiden.

Nach den Untersuchungen der Parameter werden die beiden Rekombinationsoperatoren miteinander verglichen. Führt die gerichtete Rekombination zu besseren Ergebnissen? Wird das Fitnessmaximum schneller erreicht? Wird sich immer für ein Maximum entschieden oder entstehen vielleicht mehrere gleichberechtigte Unterpopulationen?

7.1 Die Mutation

Mutationsrate und Zeichenkettenlänge Begonnen wird mit dem Einfluss des Mutationsoperators. Es ist anzunehmen, dass die Durchschnittssymmetrie der Endpopulation mit steigender Mutationsrate p_m abnimmt. Aber nicht nur eine steigende Mutationsrate sollte die Durchschnittssymmetrie senken, sondern auch eine steigende Zeichenkettenlänge l . Denn vergrößert sich l , so steigt auch die Wahrscheinlichkeit, dass eine Zeichenkette mutiert wird und damit schlimmstenfalls ihre Symmetrie gestört wird. In Abbildung 7.1 sieht man den Graphen der Durchschnittssymmetrie, zum einen bei fester Zeichenkettenlänge für zwei verschiedene Mutationsraten und zum anderen bei fester Mutationsrate und verschiedenen Zeichenkettenlängen. Die Vermutung bestätigt sich. Mit steigendem l und steigendem p_m sinkt die Symmetrie der Endpopulation.

7 Simulationsergebnisse

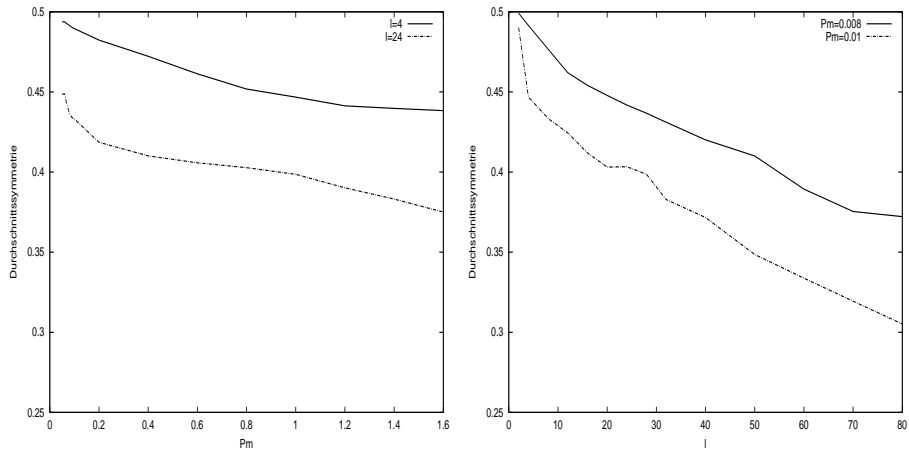


Abbildung 7.1: Durchschnittssymmetrie in Abhängigkeit von der Mutationsrate p_m in [%] (links) und der Zeichenkettenlänge l (rechts)

In Abbildung 7.2 ist die Abhängigkeit der Durchschnittssymmetrie von der Zeichenkettenlänge und der Mutationsrate in einem Graphen dargestellt. Die Werte der Symmetrie sind im rechten Graphen farbig codiert.

Alle Simulationen wurden mit einer Populationsgröße von $2n = 100$ durchgeführt und nach 50000 Generationen beendet. Jede Parameterkombination von p_m und l wurde in jeweils drei Simulationen getestet. Das sind bei je 14 Werten für p_m und l fast sechshundert Simulationen.

Endpopulation Interessant ist auch, die Zusammensetzung der Endpopulationen zu betrachten. Welche Parameterkombinationen begünstigen das Finden eines Maximums und erlauben, dass es in der Population verbleibt? Der Verbleib in der Population sollte theoretisch mit wachsenden p_m und l sinken.

In Abbildung 7.3 sieht man, dass ein Maximum ab einer Zeichenkettenlänge von 60 nicht mehr gefunden wird, dabei senkt eine Steigerung der Mutationsrate diesen Schwellenwert. Damit das Maximum in der Population erhalten bleibt und nicht immer wieder durch die Mutation zerstört wird, sollte die Wahrscheinlichkeit, dass eine Zeichenkette mutiert wird, geringer sein als 50%. Besonders zu Beginn des Algorithmus, wenn die Population nur aus wenigen Vertretern eines Maximums besteht, ist dies von besonderer Bedeutung. Es soll also gelten, dass die Wahrscheinlichkeit, mit der eine Zeichenkette nicht mutiert wird, größer als 50% ist:

$$\begin{aligned} (1 - p_m)^l &> 0.5 \\ \Rightarrow l &< \frac{\ln 0.5}{\ln (1 - p_m)}. \end{aligned}$$

Hier sieht man, dass eine Steigerung von p_m tatsächlich eine Senkung des Schwellenwertes verursacht. Konkret heißt dies, dass bei einer Mutationsrate von $p_m = 0.01$ für die Zeichenkettenlänge $l < 69$ gelten muss.

7.2 Die Rekombination

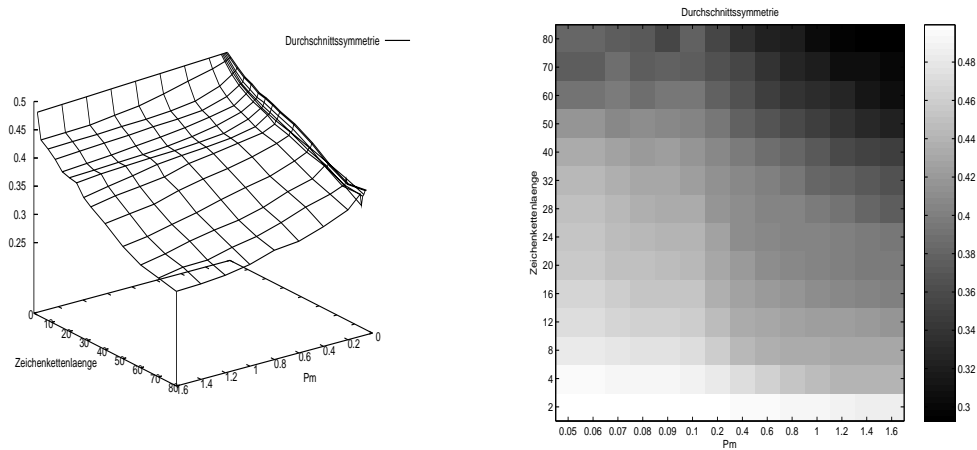


Abbildung 7.2: Durchschnittssymmetrie in Abhängigkeit von der Zeichenkettenlänge und der Mutationsrate p_m in [%]

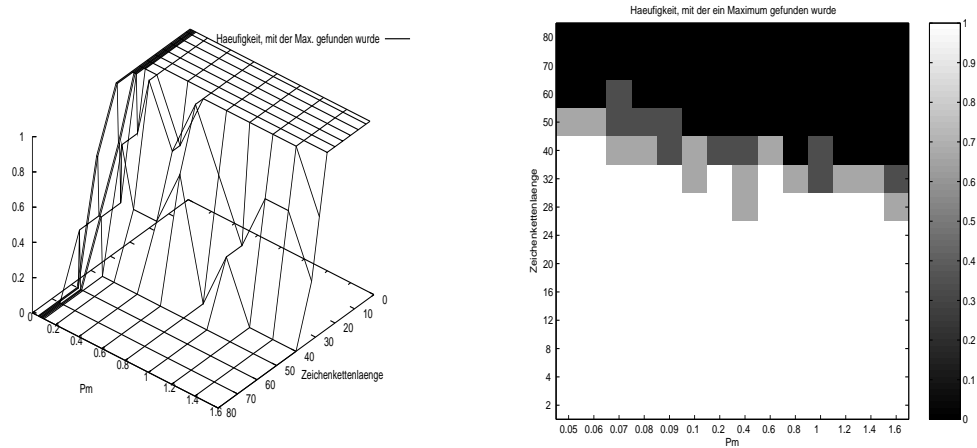


Abbildung 7.3: Häufigkeit, mit der ein Maximum in den Endpopulationen vorhanden war

Zusammenfassung Wie erwartet sinkt die Durchschnittssymmetrie, wenn die Mutationsrate bzw. die Zeichenkettenlänge erhöht wird. Die Wahrscheinlichkeit, dass ein Maximum in der Population verbleibt fällt ebenfalls wie erwartet mit steigender Mutationsrate und Zeichenkettenlänge.

7.2 Die Rekombination

Der gerichtete Rekombinationsoperator wird im Computerprogramm folgendermaßen implementiert. Es werden die zur Rekombination zugelassenen Elemente bestimmt und in einer Liste gespeichert. Dann werden zwei Elemente zufällig herausgenommen, ihre Ähnlichkeit bestimmt, das heißt die Anzahl von übereinstimmenden Bits, und entsprechend der Wahrscheinlichkeit P_{IJ}^q gekreuzt. Diese Wahrscheinlichkeit wird mit q skaliert (siehe Abschnitt 4.5). Findet keine Kreuzung statt, so wird für diese Ele-

mente vorgemerkt, dass sie eine erfolglose Kreuzung hinter sich haben.

Es gibt eine maximale Anzahl von Kreuzungsversuchen. Dieser Parameter heißt *matingTrials*. Hat ein Element diese Anzahl ausgeschöpft, so erhält es keine weitere Chance und wird aus den zur Rekombination zugelassenen Elementen entfernt.

Durch die Restriktion mit dem Parameter *matingTrials* wird ein Element bestraft, falls es keinen ähnlichen Partner findet, indem sein Nachwuchs ausbleibt und seine Gene nicht in die nächste Generation übernommen werden.

Wurden nach diesem Verfahren nicht n Paare gefunden, so werden die restlichen Plätze mit bereits gefundenen Paaren belegt. Diese Paare erhalten also statt zwei Nachkommen vier oder sogar mehr Nachkommen. Wurde bei dem Verfahren gar kein Paar gefunden, das heißt die Eltern waren sich nicht hinreichend ähnlich oder es wurden nur unglückliche Zusammensetzungen als Paare getestet, so wird die normale Rekombination durchgeführt und zwar mit den Individuen, die zur Rekombination ausgewählt wurden. Die normale Rekombination wird also als Strafe verwendet. Dabei ist es fraglich, inwieweit eine normale Kreuzung die Fitness der Nachkommen senkt oder sie sich nur noch verbessern kann, wenn sich kein hinreichend ähnliches Paar finden lässt.

Solche Anpassungen sind vonnöten, da im Computerprogramm einerseits mit einer endlichen Populationsgröße gearbeitet wird und andererseits nur begrenzte Rechnerkapazitäten zur Verfügung stehen. So kann und möchte man nicht unbegrenzt lange warten, bis sich ausreichend ähnliche Elternpaare gefunden haben und führt sinnvolle Begrenzungen, wie den Parameter *matingTrials* oder ein Ausweichverfahren, wie die normale Rekombination ein.

Alle Simulationen in diesem Abschnitt verwenden eine Populationsgröße von $2n = 500$ und eine Zeichenkettenlänge von $l = 40$. Die Mutationsrate ist $p_m = 0.01$ und die Rekombinationsrate $p_c = 0.9$. Jede Parameterkombination wurde in je drei Simulationen getestet.

Vergleich der beiden Rekombinationsoperatoren Zunächst soll die Durchschnittssymmetrie der Endpopulationen verglichen werden, wenn der normale Rekombinationsoperator und der gerichtete Operator verwendet wurde.

In Abbildung 7.4 sieht man, dass der gerichtete Operator nur für bestimmte q -Werte ein besseres Ergebnis liefert als die normale Rekombination. Ist q zu klein oder zu groß, so sorgt die normale Rekombination für eine fittere Endpopulation. Liegt aber q in einem bestimmten Bereich, der in diesem Beispiel von 4 bis 10 reicht, so liefert die gerichtete Rekombination bessere Ergebnisse. Dies ist folgendermaßen zu erklären: Verwendet man die gerichtete Rekombination, so steigt mit wachsenden q der Druck auf ein Individuum, einen ähnlichen Partner zu finden. Isolierte Individuen werden somit schneller aussortiert. Je ähnlicher die zu kreuzenden Partner sind, umso wahrscheinlicher ist es, dass ihre Nachkommen ungefähr die

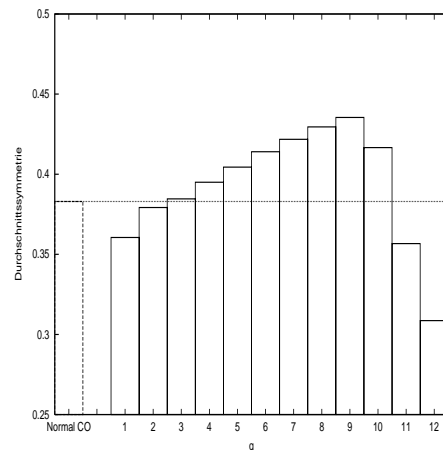


Abbildung 7.4: Vergleich normale Rekombination vs. gerichtete Rekombination

gleichen Fitnesswerte besitzen. Bei der gerichteten Rekombination bleiben die Fitnesswerte der Nachkommen von fitten Individuen hoch, wohingegen die normale Rekombination aus fitten Eltern unfitte Nachkommen erzeugen kann.

Die gerichtete Rekombination unterbindet mehr Kreuzungen als die normale Rekombination. Dies ist besonders hinderlich, wenn noch keine symmetrischen Individuen gefunden worden sind, wie es zum Beispiel zu Beginn des Genetischen Algorithmus der Fall ist. Die Kreuzung zweier unsymmetrischer Partner kann durchaus von Vorteil sein. Durch diese geringere Kreuzungsfrequenz schneidet die gerichtete Rekombination für kleine q schlechter ab als die normale Rekombination. Ist q jedoch größer, so kompensiert die gerichtete Rekombination die geringere Kreuzungsfrequenz durch die Kreuzung von ähnlicheren Individuen und sorgt damit für ein besseres Ergebnis, während die normale Rekombination immer wieder symmetrische Individuen zerstört.

Als Konsequenz sollte zu Beginn des Genetischen Algorithmus der Fitnessanstieg bei Benutzung der normalen Rekombination größer sein als bei Benutzung der gerichteten Rekombination. In Abbildung 7.5 wird dies bestätigt. Für die Praxis kann man diese Erkenntnis nutzen, indem man beide Rekombinationsverfahren kombiniert. Zu Beginn des Genetischen Algorithmus ist die Verwendung der normalen Rekombination zu empfehlen und nach einer bestimmten Generationenzahl die gerichtete Rekombination. So sollte der Genetische Algorithmus schneller die Gleichgewichtspopulation erreichen und eine fittere Endpopulation liefern. Damit könnten zum Beispiel die Simulationen in Abbildung 7.5 um circa 5000 Generationen verkürzt werden.

Der Leistungsabfall für hinreichend große q hängt mit dem Parameter *matingTrials* zusammen, der in diesen Simulationen konstant geblieben ist. Es existiert für die Parameterkombination von q und *matingTrials* eine Schwelle, an der die Leistung des Genetischen Algorithmus stark abfällt. Diese Schwelle wird im folgenden Paragraphen näher untersucht.

7 Simulationsergebnisse

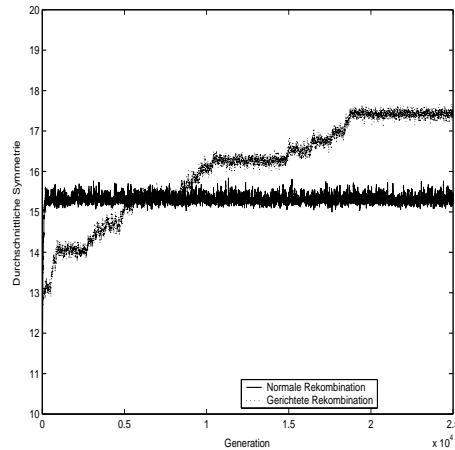


Abbildung 7.5: Vergleich der beiden Rekombinationsverfahren bzgl. des Symmetrieanstiegs in der Population. Zeichenkettenlänge ist $l = 40$.

Zunächst wird das Interesse aber auf die Homogenität der Endpopulationen gelenkt. Es soll untersucht werden, ob der gerichtete Operator tatsächlich eine bezüglich des Erbguts homogenere Endpopulation hervorruft als die normale Rekombination.

In Abbildung 7.6 sind die Ergebnisse der Endpopulationen abgetragen. Zum einen wurde die Anzahl an Clustern ermittelt und zum anderen die Clustergröße. Ein Cluster ist eine Menge von Individuen in der Population, die denselben Genotyp besitzen. Dabei werden nur Cluster gezählt, die wenigstens fünf Vertreter haben. Es ist zu erkennen, dass mit der gerichteten Rekombination wesentlich mehr Cluster in der Population gebildet werden und dass die Anzahl an Clustern mit q ansteigt. Dies ist ein Resultat des mit q steigenden Drucks, ähnliche Partner in der Rekombination zu finden.

In der rechten Graphik von Abbildung 7.6 ist die durchschnittliche Anzahl an Individuen pro Cluster abgetragen. Hier ist kein eindeutiger Trend zu erkennen, aber man sieht, dass die Clustergröße bei beiden Rekombinationsarten ungefähr gleich groß ist.

Die höhere Clusteranzahl bei ähnlicher Clustergröße zeigt, dass die gerichtete Rekombination eine homogenere Endpopulation erzeugt als die normale Rekombination. Der Grad der Homogenisierung steigt mit q . Vereinen die Cluster fitte Individuen, so ist damit auch die fittere Endpopulation bei Verwendung der gerichteten Rekombination zu erklären.

Die gerichtete Rekombination Es wurde bereits erwähnt, dass die Parameter *matingTrials* und q nicht trivial zusammenhängen. Wie in Abbildung 7.7 zu sehen ist, nimmt die Durchschnittssymmetrie mit steigenden q und sinkenden *matingTrials* leicht zu, ab einer gewissen Schwelle aber fällt die Leistung des Genetischen Algorithmus ab.

Die Existenz dieser Schwelle ist leicht begründet. Je größer q wird, um-

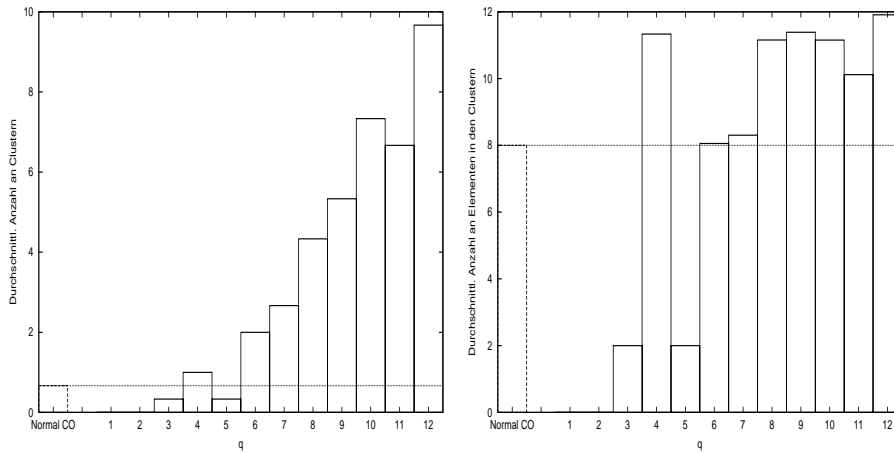


Abbildung 7.6: Vergleich der normalen Rekombination (NormalCO) mit gerichteter Rekombination bezüglich der durchschnittlichen Anzahl von Clustern (links) und durchschnittlicher Clustergröße (rechts)

so schwieriger ist es selbst für stark ähnliche Paare gekreuzt zu werden, sofern sie sich in wenigstens einem Bit unterscheiden. Wird der Parameter *matingTrials* zu niedrig gesetzt, das heißt die Individuen haben weniger Versuche, einen ähnlichen Partner zu finden, so werden keine ähnlichen Eltern gefunden, die normale Kreuzung findet statt und zerstört die fitten Individuen.

Die Durchschnittssymmetrie steigt mit wachsenden q und sinkenden *matingTrials*, da durch sie der Kreuzungsdruck erhöht wird. Je größer q , umso ähnlicher müssen sich Eltern sein, um gekreuzt zu werden und umso besser ist dann das Kreuzungsergebnis, da der Fitnesswert der Nachkommen im Bereich der Eltern liegt. Je geringer *matingTrials*, umso weniger Versuche haben die Eltern zur Verfügung einen ähnlichen Partner zu finden und umso schneller werden isolierte Eltern aussortiert und die Nachkommenstreuung in der Fitnesslandschaft verringert.

Dieses Verhalten ist auch analytisch zu erklären. Die Wahrscheinlichkeit, dass ein Element I wegen mangelnden ähnlichen Partnern aussortiert wird, ist:

$$P(I \text{ wird aussortiert}) = P'(I) \left(1 - \sum_J P'(J) P_{IJ}^q \right)^{mT}. \quad (7.1)$$

Dabei wird der Parameter *matingTrials* mit mT abgekürzt. Die Wahrscheinlichkeit $P(I \text{ wird aussortiert})$ sinkt mit wachsenden P_{IJ} bzw. mit fallenden q und wachsenden mT . Erhöht man also q oder senkt die Anzahl der Versuche, einen Partner zu finden, mT , so steigt die Wahrscheinlichkeit, dass eine Zeichenkette I keinen ausreichend ähnlichen Partner findet. Damit wird der Rekombinationsdruck erhöht, das heißt nur sehr ähnliche

7 Simulationsergebnisse

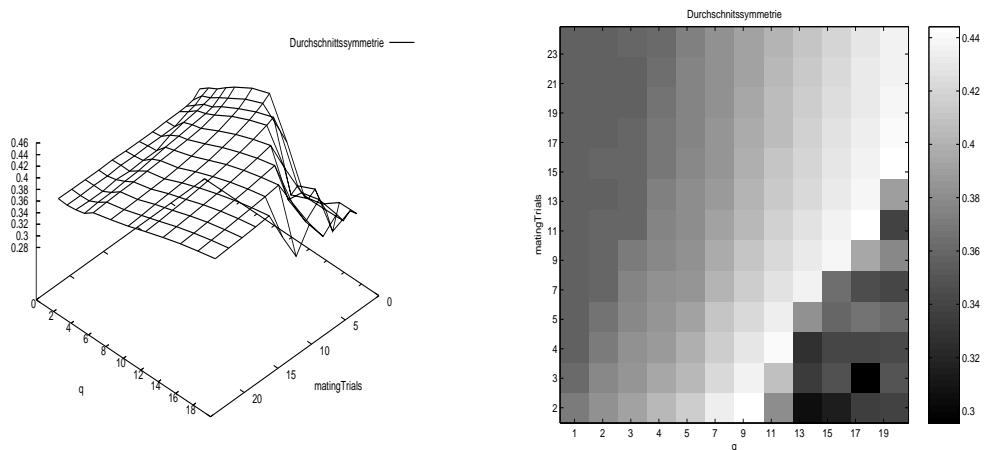


Abbildung 7.7: Durchschnittssymmetrie in Abhängigkeit von q und $matingTrials$

Partner kreuzen sich und isolierte Individuen werden aussortiert. Bis zu einem bestimmten Punkt sollte sich also die Leistung des Genetischen Algorithmus verbessern, dann aber abrupt abfallen, da sich bei zu kleinem mT bzw. bei zu großem q keine Paare mehr finden können. Die fitten Individuen werden dann durch die normale Rekombination zerstört. Bestätigt wird dies durch einen Blick auf Abbildung 7.8, in der die Nutzung der normalen Rekombination, wenn sich bei der gerichteten Rekombination keine Paare gefunden haben, eingetragen ist. Tatsächlich steigt die Häufigkeit der Benutzung der normalen Rekombination mit wachsenden q und fallenden mT an. In einer nicht zufälligen, sondern eher geordneten Population muss somit notwendigerweise durch die willkürliche Elternwahl die Leistung des Genetischen Algorithmus abfallen, das heißt die Durchschnittssymmetrie der Endpopulationen sinken.

Ein weiteres Indiz, dass ein großes q den Druck ähnliche Partner für die Rekombination zu finden steigert, müsste eine wachsende Anzahl an gleichen Individuen in der Population sein. In Abbildung 7.9 ist die durchschnittliche Clustergröße abgetragen. Es ist zu erkennen, dass die Clustergröße mit q steigt. Je größer q , desto mehr Vertreter gleichen Genotyps existieren. Dies ist ein Resultat des erhöhten Drucks, einen ähnlichen Partner zu finden.

Endpopulation Nun sollen die Endpopulationen betrachtet werden. In Abbildung 7.10 ist zu erkennen, dass ein Maximum so gut wie immer gefunden wird, wenn q und $matingTrials$ nicht die kritische Schwelle überschreiten, ab der die Leistung des Algorithmus abfällt.

In Abbildung 7.11 sieht man, dass sich die Anzahl der Cluster zwischen 0 und 8 bewegt und abrupt ansteigt, wenn die kritische Schwellenkombination von q und $matingTrials$ überschritten wird. Man kann also vermuten, dass der Genetische Algorithmus mehr Cluster bildet, wenn das Optimum noch nicht gefunden wurde, bzw. wenn die Population durch die

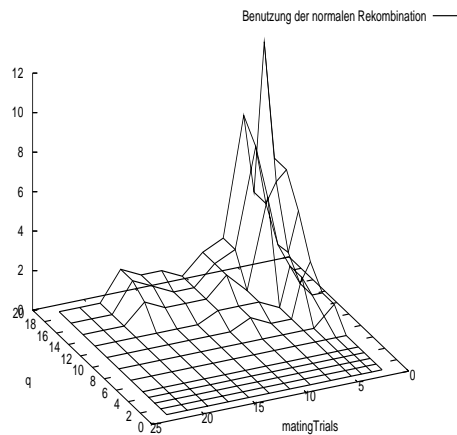


Abbildung 7.8: Häufigkeit, mit der normale Rekombination benutzt wurde, in Abhängigkeit von q und $matingTrials$

Benutzung der normalen Rekombination noch stark über dem Suchraum verstreut ist.

Eine weitere Erkenntnis lässt sich aus Abbildung 7.11 ziehen. Da die Anzahl der Cluster mit wachsenden q und fallenden $matingTrials$ steigt, begünstigt ein großes q und ein kleines $matingTrials$ die Artenbildung. Fasst man Individuen gleichen Genotyps zu einer Art zusammen, so bedeuten mehrere Cluster mehrere Arten. Es war zu hoffen, dass die gerichtete Rekombination zur Artenbildung beiträgt. Dies hat sich nicht bestätigt, da sich die Cluster in diesen Simulationen fast ausschließlich in der Umgebung eines Maximums befanden und sich nicht um mehrere Maxima gruppierten.

Zusammenfassung Die Durchschnittssymmetrie der Endpopulation bei Verwendung der gerichteten Rekombination ist wesentlich höher, jedoch wird die Gleichgewichtspopulation wesentlich langsamer erreicht. Es wird die Kombination der beiden Rekombinationsverfahren vorgeschlagen.

Eine geeignete Wahl der Parameter q und $matingTrials$ erhöht die Durchschnittssymmetrie. Wird $matingTrials$ jedoch zu klein für q gesetzt, sinkt die Durchschnittssymmetrie unter den Wert, der mit der normalen Rekombination erreicht wird.

7.3 Die Fitnessfunktion

Alle Simulationen in diesem Abschnitt verwenden eine Populationsgröße von $2n = 500$ und eine Zeichenkettenlänge von $l = 40$. Die Mutationsrate ist $p_m = 0.01$ und die Rekombinationsrate $p_c = 0.9$. Der Parameter q wurde auf 2 gesetzt.

7 Simulationsergebnisse

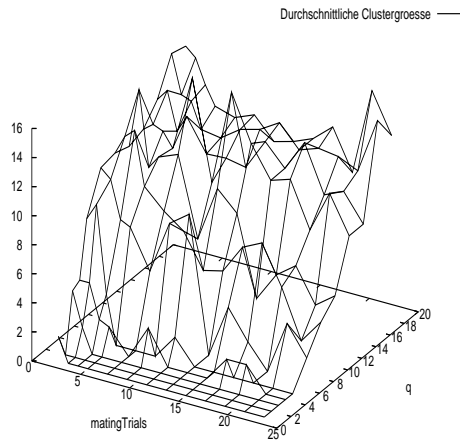


Abbildung 7.9: Durchschnittliche Clustergröße in Abhängigkeit von q und $matingTrials$

Bisher wurde eine lineare Fitnessfunktion verwendet. Die Anzahl symmetrischer Bits wurde gemessen und ergab den Fitnesswert eines Elementes. Dieser Fitnesswert lag zwischen 0 und $\frac{1}{2}$. Wird die Fitnessfunktion durch eine Potenz p nichtlinearisiert, also f_t durch f_t^p ersetzt, so erzielt man eine bessere Leistung des Genetischen Algorithmus. In Abbildung 7.12 sieht man, dass mit wachsenden p auch die Durchschnittssymmetrie der Endpopulation steigt. Mit p kontrolliert man den Selektionsdruck. Je größer p , umso resoluter werden weniger fitte, also weniger symmetrische Individuen aussortiert.

In Abbildung 7.13 ist die Durchschnittssymmetrie in Abhängigkeit von p und q abgetragen. Wie bereits gesehen wurde, steigt die Symmetrie mit p und q , wobei die Steigerung in q -Richtung eher gering ist. In den Abschnitten 3.5 und 4.6 wurde gezeigt, dass die Selektion und die Rekombination gemeinsam die Homogenisierung der Population fördern. Verstärkt man nun den Selektionsdruck sowie den Rekombinationsdruck, so sollte die Homogenisierung schneller vonstatten gehen. In Abbildung 7.14 ist zu erkennen, dass der Fitnessanstieg stärker und somit auch die Homogenisierung mit großen p schneller ist. Die Daten sind gemittelt aus drei Simulationen, bei denen q die Werte 7, 9 und 11 zugewiesen wurden.

Dass die Parameter p und q nicht nur für eine schnellere Homogenisierung, sondern auch für eine insgesamt homogenere Population sorgen, sieht man in Abbildung 7.15. Die Anzahl an Individuen in einem Cluster steigt mit p und q , wobei auch hier die größte Steigerung in p -Richtung stattfindet. Mit dem Selektionsdruck kann man also nicht nur die Homogenisierungsgeschwindigkeit, sondern auch den Grad der Homogenisierung beeinflussen.

7.3 Die Fitnessfunktion

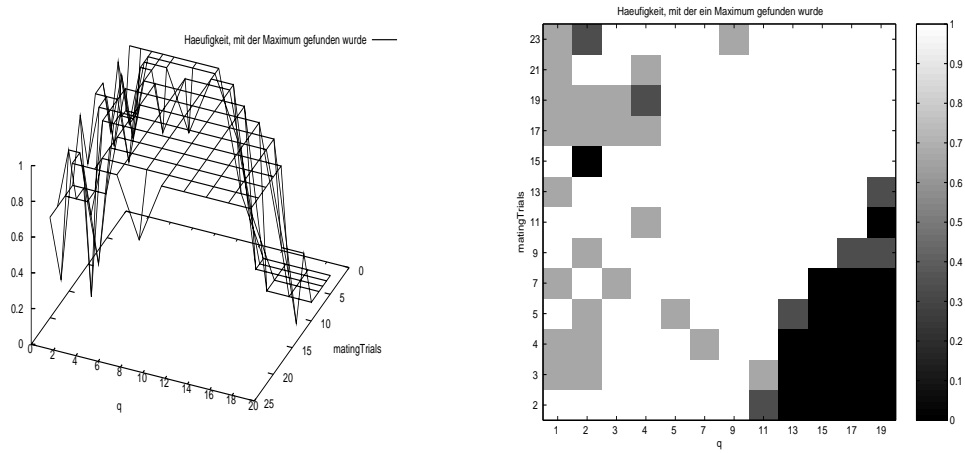


Abbildung 7.10: Häufigkeit, mit der ein Maximum gefunden wurde

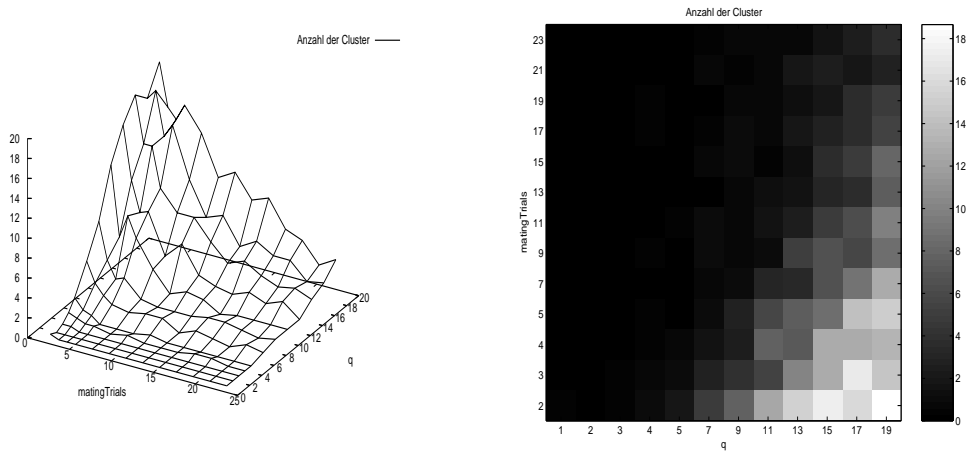


Abbildung 7.11: Anzahl der Cluster in Abhängigkeit von q und $matingTrials$

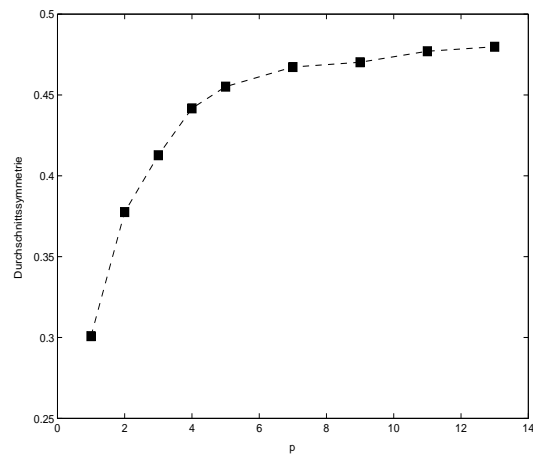


Abbildung 7.12: Durchschnittssymmetrie in Abhängigkeit von p

7 Simulationsergebnisse

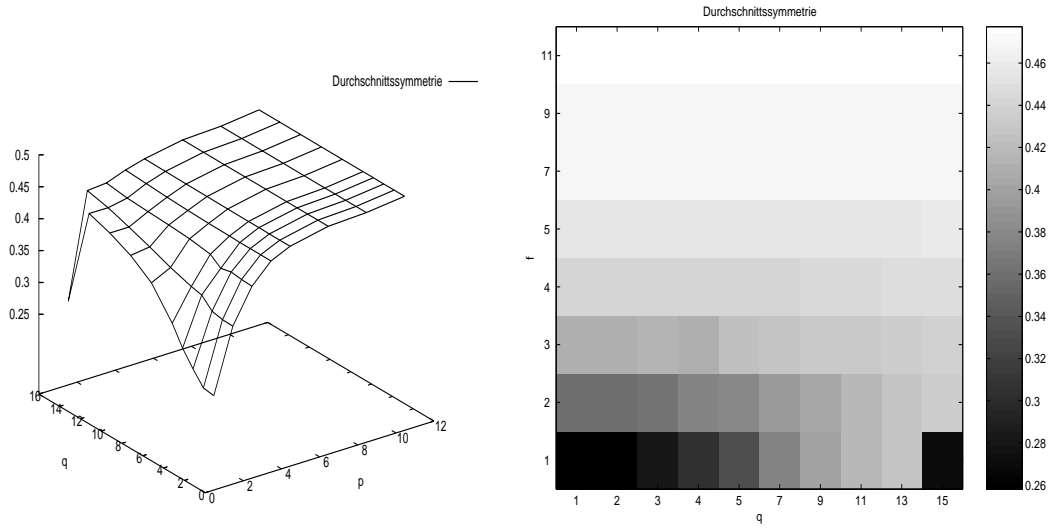


Abbildung 7.13: Durchschnittssymmetrie in Abhängigkeit von p und q

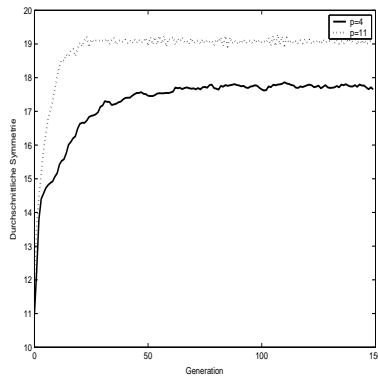


Abbildung 7.14: Fitnessanstieg für $q = 7, 9, 11$ und Zeichenkettenlänge $l = 40$

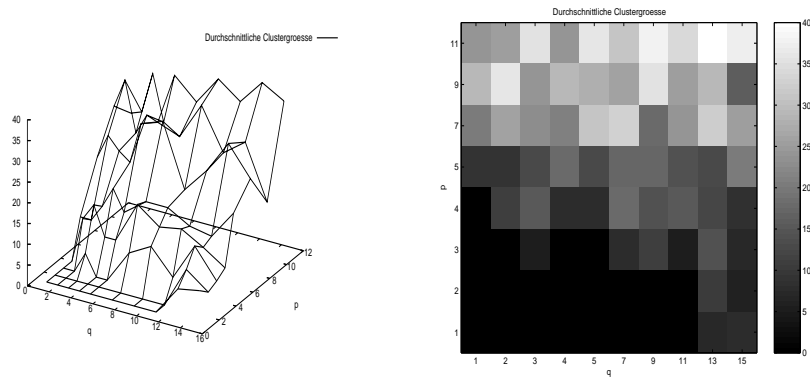


Abbildung 7.15: Durchschnittliche Clustergröße

8 Zusammenfassung

Im Folgenden werden die Leistungen, die für diese Diplomarbeit erbracht wurden nochmals aufgeführt.

Die Arbeit befasste sich mit der theoretischen Behandlung von Genetischen Algorithmen. Ein Kernpunkt bildete die Formalisierung der genetischen Operatoren Selektion, Rekombination und Mutation.

Ein wesentlicher Aspekt war die Aufstellung einer Evolutionsgleichung, die die Ausbreitung von Schemen in der Population quantifiziert.

Die Untersuchungen wurden auf Genetische Algorithmen ausgeweitet, die eine Symmetrie-Fitnessfunktion und einen gerichteten Rekombinationsoperator verwenden. Die Wirkung der genetischen Operatoren auf der neuen Fitnesslandschaft wurde analysiert und eine angepasste Evolutionsgleichung hergeleitet.

Theoretische Aspekte wie die Effektive Fitness, die String- und Bausteinbasis, die Tensorarstellung der genetischen Operatoren sowie die Rolle der Renormierungsgruppe bei der Lösung der Evolutionsgleichung wurden erläutert.

Den Abschluss bildeten numerische Untersuchungen, die die Leistung des Genetischen Algorithmus bezüglich verschiedener Parameterbelegungen testeten und die Auswirkungen des gerichteten Rekombinationsoperators beleuchteten. Wesentliches Ergebnis ist, dass die Endpopulation bei der Verwendung der gerichteten Rekombination zwar sehr viel langsamer den Gleichgewichtszustand erreicht, ihre Elemente jedoch erheblich fitter sind als bei der Verwendung der normalen Rekombination.

Ein weiteres Resultat ist, dass mit dem Selektions- und Rekombinationsdruck der Grad und die Geschwindigkeit der Homogenisierung des Erbguts in der Population gesteuert werden kann.

8 Zusammenfassung

Bezeichnungen

$ M $	Mächtigkeit einer Menge M .
$ x \wedge y $	Anzahl der Einsen, an denen die Zeichenketten x und y übereinstimmen.
$(\Omega, \mathfrak{D}, P)$	Wahrscheinlichkeitsraum über Ω . Dabei ist \mathfrak{D} eine σ -Algebra und P ein Wahrscheinlichkeitsmaß.
*	Symbol für den Fall, dass der Wert eines Detektors irrelevant ist.
\otimes	Tensorprodukt.
\mathfrak{A}	Suchraum. Die Menge aller möglichen Lösungen eines Optimierungsproblems.
$\hat{\mathfrak{A}}^n$	Ungeordnete n -elementige Menge von Elementen aus \mathfrak{A} . Mehrmaliges Auftreten eines Elementes ist möglich.
$(\hat{\mathfrak{A}}^2)^n$	Menge aller n -elementigen Teilmengen von $\hat{\mathfrak{A}}^2$.
B_S	Stringbasis.
B_T	Taylor-Basis.
B_V	Vertex-Basis.
B_W	Walsh-Basis.
B_x	Bausteinbasis zum Bezugspunkt x .
B_x -Träger	Menge der Schemen, die x enthalten.
$CF(\mathfrak{A})$	Raum der charakteristischen Funktionen über \mathfrak{A} .
Δ	Menge der Detektoren.
$\delta_i : \mathfrak{A} \rightarrow V_i$	Detektor mit Wertebereich V_i .
δ_{ij}	Kronecker Delta
$\delta(H)$	Länge des Schemas H . Differenz zwischen der ersten und der letzten fixierten Stelle von H .
$D(z)$	Definitionsbereich einer Funktion z .
d_{IJ}^H	Hamming-Abstand zwischen den Zeichenketten I und J .
EA	Erwartungswert einer Zufallsgröße A .
E_N	N -dimensionale Einheitsmatrix.
E_S	Erwartete Anzahl an Vertretern eines Schemas in der Folgepopulation.
f_{eff}	Effektive Fitness. Gibt den Fortpflanzungserfolg eines Populationselementes an.
f_c	Wahrscheinlichkeitsmaß für die Rekombination.
f_m	Wahrscheinlichkeitsmaß für die Mutation.
$f_t(x) = \frac{F(x)}{\sum_{y \in P_t} F(y)}$	Relative Fitness des Individuums $x \in P_t$.

Bezeichnungen

$f_t(H) = \sum_{x \in P_t \cap H} f_t(x)$	Relative Fitness des Schemas H in P_t .
F	Selektionstensor, der die Selektion bezüglich der Stringbasis beschreibt.
F_B	Selektionstensor bezüglich der Bausteinbasis.
F_W	Selektionstensor bezüglich der Walsh-Basis.
$F(x)$	Fitnesswert des Elements $x \in \mathfrak{A}$.
$\bar{F}_t = \frac{1}{2^n} \sum_{x \in P_t} F(x)$	Mittlere Fitness der Population P_t .
$F_t(H) = \frac{1}{m(H,t)} \sum_{x \in P_t \cap H} F(x)$	Mittlere Fitness des Schemas H in P_t .
$Fun(\mathfrak{A})$	Raum der reellwertigen Funktionen über \mathfrak{A} .
$Fun^*(\mathfrak{A})$	Dualraum von $Fun(\mathfrak{A})$. Raum der linearen Funktionale über \mathfrak{A} .
$h_1, \dots, h_{\delta(H)}$	Zeichenzwischenräume des Schemas H .
$H = (s_1, \dots, s_m)$	Schema. Menge von Elementen $x \in \mathfrak{A}$, für die gilt: $\delta_i(x) = s_i$ falls $s_i \neq * \forall i = 1, \dots, m$.
H_k	Schema, dessen fixierte Positionen und dessen Allele an diesen Positionen mit denen von H bis zur Stelle $k \in \{h_1, \dots, h_{\delta(H)}\}$ übereinstimmen.
\hat{H}_k	Schema, dessen fixierte Positionen und dessen Allele an diesen Positionen mit denen von H von der Stelle $k \in \{h_1, \dots, h_{\delta(H)}\}$ an übereinstimmen.
\bar{H}_i	Klasse von Schemen, deren Allele an den fixierten Positionen an genau i Positionen verschieden von denen von H sind.
H_L	Schema, das mit H an den Stellen übereinstimmt, an denen die Maskenbits einer Rekombinationsmaske 1 sind.
H_R	Schema, das mit H an den Stellen übereinstimmt, an denen die Maskenbits einer Rekombinationsmaske 0 sind.
$\lambda_I^{JK}(M)$	Komponente des Rekombinationstensors λ . Der Wert von $\lambda_I^{JK}(M)$ ist genau dann 1, wenn der erste Nachkomme der Eltern J und K unter Verwendung der Rekombinationsmaske M gleich I ist.
λ_I	Komponente des Rekombinationstensors bei festem I . Der Tensor ist bezüglich der Stringbasis gegeben.
λ_I^B	Komponente des Rekombinationstensors bei festem I . Der Tensor ist bezüglich der Bausteinbasis gegeben.
Λ	Transformationsmatrix von der Stringbasis zur Bausteinbasis bei einer Zeichenkettenlänge von $l = 1$.
Λ^l	Transformationsmatrix von der Stringbasis zur Bausteinbasis bei einer Zeichenkettenlänge von l .
Λ_0^1	Transformationsmatrix von der Bausteinbasis B_0 zur Bausteinbasis B_1 . Zeichenkettenlänge ist 1.

Λ_W	Transformationsmatrix von der Stringbasis zur Walsh-Basis bei einer Zeichenkettenlänge von $l = 1$.
Λ_W^l	Transformationsmatrix von der Stringbasis zur Walsh-Basis bei einer Zeichenkettenlänge von l .
l	Zeichenketten- bzw. Genomlänge und Anzahl der Detektoren.
$m(H, t) = P_t \cap H $	Anzahl der Vertreter von H in P_t .
$mT, \text{ matingTrials}$	Maximale Anzahl der Versuche, die ein Individuum hat, um einen Partner bei der gerichteten Rekombination zu finden.
$M \setminus N$	Menge der Elemente aus M , die nicht in N liegen.
M	Tensor, der die Mutation bezüglich der Stringbasis beschreibt.
M	Rekombinationsmaske.
M_B	Mutationstensor bezüglich der Bausteinbasis.
M_B^l	Matrix der Bausteinbasisfunktionen. Die (i, j) -te Komponente gibt den Funktionswert des j -ten Elementes an der i -ten Basisfunktion wieder.
M_c	Menge aller Rekombinationsmasken.
M_l	Matrix, dessen (i, j) -te Komponente angibt, mit welcher Wahrscheinlichkeit das j -te Element des Suchraumes in das i -te Element mutiert wird.
M_S^l	Matrix der Stringbasisfunktionen. Die (i, j) -te Komponente gibt den Funktionswert des j -ten Elementes an der i -ten Basisfunktion wieder.
M_W	Mutationstensor bezüglich der Walsh-Basis.
$\Xi = \prod_{i=1}^m \{V_i \cup \{\star\}\}$	Menge aller Schemen.
$2n$	Populationsgröße.
\mathbb{N}	Menger der natürlichen Zahlen.
$N = 2^l$	Anzahl der Elemente des Suchraums.
$o(H)$	Ordnung des Schemas H . Anzahl der fixierten Positionen von H .
O_K	Kreuztauschoperator.
O_M	Mutationsoperator.
O_S	Selektionsoperator.
$\mathfrak{P}(M)$	Potenzmenge von M .
\mathbb{P}	Menge aller möglichen Populationen.
$p(M)$	Wahrscheinlichkeit, dass Rekombinationsmaske M verwendet wird.
p_c	Kreuztauschwahrscheinlichkeit.
p_m	Mutationswahrscheinlichkeit.
$P(A)$	Wahrscheinlichkeit eines Ereignisses A .
$P(A/B)$	Bedingte Wahrscheinlichkeit. Die Wahrscheinlichkeit des Ereignisses A unter der Bedingung, dass B eintritt.

Bezeichnungen

$P(H, t)$	Wahrscheinlichkeit, dass Vertreter des Schemas H in der Population P_t auftreten.
$P(I, t)$	Wahrscheinlichkeit, dass Zeichenkette I in der Population P_t auftritt.
P	Populationsvektor bezüglich der Stringbasis. Die i -te Komponente gibt die relative Häufigkeit des i -ten Elementes des Suchraums in der Population wieder.
P'	Vektor, dessen Komponenten die relativen Häufigkeiten der Elemente nach der Selektion beschreiben.
P_B	Populationsvektor bezüglich der Bausteinbasis.
P_c	Vektor, dessen Komponenten die relativen Häufigkeiten der Elemente nach der Rekombination beschreiben.
P_{gc}	Vektor, dessen Komponenten die relativen Häufigkeiten der Elemente nach der gerichteten Rekombination beschreiben.
P_{IJ}^q	Anzahl der Bits, an denen die Zeichenketten I und J übereinstimmen, potenziert mit q .
P_m	Vektor, dessen Komponenten die relativen Häufigkeiten der Elemente nach der Mutation beschreiben.
$P_m(I \rightarrow J)$	Wahrscheinlichkeit, mit der Zeichenkette I in Zeichenkette J mutiert wird.
$P_m(H_a \rightarrow H_b)$	Wahrscheinlichkeit, mit der ein Vertreter von H_a in einen Vertreter von H_b mutiert wird.
$P_t \in \mathbb{P}$	Population der Generation t ($t \in \mathbb{N}_0$). Eine endliche Menge von Elementen des Suchraumes. Es gilt $P_t \not\subseteq \mathfrak{A}$, da ein $x \in \mathfrak{A}$ mehrmals in P_t auftreten kann.
P_W	Populationsvektor bezüglich der Walsh-Basis.
$\mathbb{R}[a, b]$	Menge der reellen Zahlen, die im Intervall von a bis b liegen.
s	Anzahl der symmetrischen Bits in einer Zeichenkette.
u	Anzahl der unsymmetrischen Bits in einer Zeichenkette.
xc_i	Element des Suchraums der Dimension $l + 1$, wenn c_i ein Element des Suchraums der Dimension l ist. Das Element xc_i wird gebildet, indem $x \in \{0, 1\}$ von vorn an c_i angefügt wird.
xb_i	Baustein des Suchraums der Dimension $l + 1$, wenn c_i ein Baustein des Suchraums der Dimension l ist. Der Baustein xb_i wird gebildet, indem $x \in \{0, 1, *\}$ von vorn an b_i angefügt wird.
yx	Menge, die aus den Elementen des Suchraums der Dimension $l + 1$ besteht, deren Allel des ersten Bits ein y ist. Die Variable x durchläuft alle Elemente des Suchraums der Dimension l .

Literaturverzeichnis

- [Alt95] ALTENBERG, LEE: *The schema theorem and price's theorem*. In WHITLEY, DARRELL and MICHAEL VOSE (editors): *Foundations of Genetic Algorithms 3*. Morgan Kaufmann, San Francisco, 1995.
- [Bag67] BAGLEY, J.D.: *The behavior of adaptive systems which employ genetic and correlation algorithms*. PhD thesis, University of Michigan, 1967. Dissertation Abstracts International, 28(12).
- [CS] CHRYSOMALAKOS, CHRYSOMALIS and CHRISTOPHER R. STEPHENS: *What basis for genetic dynamics?* In *Proceedings of GECCO 2004*, pages 1018–1029, Berlin. Springer.
- [FHM91] FORREST, STEPHANIE, JOHN H. HOLLAND, and MELANIE MITCHELL: *The royal road for genetic algorithms: Fitness landscapes and GA performance*. In *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, 1991.
- [Fog99] FOGEL, LAWRENCE J.: *Intelligence through simulated evolution: forty years of evolutionary programming*. John Wiley, New York, 1999.
- [FOW66] FOGEL, LAWRENCE.J., A.J. OWENS, and M.J. WALSH: *Artificial Intelligence through Simulated Evolution*. John Wiley, New York, 1966.
- [Gol89] GOLDBERG, DAVID E.: *Genetic Algorithms in Search, Optimization, and Machine learning*. Addison Wesley, Reading Massachusetts, 1989.
- [Gol91] GOLDBERG, DAVID E.: *Real coded genetic algorithms, virtual alphabets, and blocking*. In *Complex Systems 5*, pages 139–167. 1991.
- [GZZZ95] GROSCHE, G., E. ZEIDLER, D. ZIEGLER und V. ZIEGLER: *Teubner-Taschenbuch der Mathematik*. Teubner, Stuttgart, Leipzig, 1995.
- [Hae96] HAEFNER, JAMES W.: *Modeling biological systems: principles and applications*. Chapman & Hall, New York, 1996.

Literaturverzeichnis

- [Hil01] HILLERMEIER, CLAU: *Nonlinear multiobjective optimization: a generalized homotopy approach*, volume 135 of *International series of numerical mathematics*. Birkhäuser, 2001.
- [Hol75] HOLLAND, JOHN H.: *Adaption in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. MIT Press, Cambridge Massachusetts, 1975.
- [Koz92] KOZA, JOHN R.: *Genetic programming: On the programming of computers by means of natural selection*. MIT Press, Cambridge Massachusetts, 1992.
- [Mic96] MICHALEWICZ, ZBIGNIEW: *Genetic algorithms + data structures = evolution programs*. Springer-Verlag, New York, 1996.
- [Mit96] MITCHELL, MELANIE: *An introduction to genetic algorithms*. MIT Press, Cambridge Massachusetts, 1996.
- [Nis97] NISSEN, VOLKER: *Einführung in Evolutionäre Algorithmen: Optimierung nach dem Vorbild der Evolution*. Vieweg, Braunschweig, Wiesbaden, 1997.
- [Poh00] POHLHEIM, HARTMUT: *Evolutionäre Algorithmen: Verfahren, Operatoren und Hinweise für die Praxis*. Springer-Verlag, Berlin, Heidelberg, 2000.
- [Rec73] RECHENBERG, I.: *Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog, Stuttgart, 1973.
- [Ree93] REEVES, COLIN R.: *Using genetic algorithms with small populations*. In FORREST, S. (editor): *Proceedings of the Fifth International Conference on Genetic Algorithms*, San Mateo California, 1993. Morgan Kaufmann.
- [SH94] SCHÖNEBURG, EBERHARD und FEDDERSEN HEINZMANN: *Genetische Algorithmen und Evolutionsstrategien: Eine Einführung in die Theorie und Praxis der simulierten Evolution*. Addison-Wesley, Bonn, Paris, Reading Mass., 1994.
- [SH98] STEPHENS, CHRISTOPHER R. and WAELBROECK H.: *Analysis of the effective degrees of freedom in genetic algorithms*. Physical Review, D57:3251–3264, 1998. adap-org/9611005.
- [SS02] STADLER, PETER F. and CHRISTOPHER R. STEPHENS: *Landscapes and effective fitness*. 2002. citeseer.nj.nec.com/596036.html.
- [Ste] STEPHENS, CHRISTOPHER R. Private Korrespondenz.

- [Ste98] STEPHENS, CHRISTOPHER R.: *Effect of mutation and recombination on the genotype-phenotype map*. In *Proceedings of GECCO 99*, pages 1382–1390, 1998. nlin.AO/0006051.
- [Ste99] STEPHENS, CHRISTOPHER R.: *“Effective” fitness landscapes for evolutionary systems*. In *Proceedings of CEC99*, pages 703–714. IEEE Press, 1999. nlin.AO/0006050.
- [Ste01] STEPHENS, CHRISTOPHER R.: *Some exact results from a coarse grained formulation of genetic dynamics*. In *Proceedings of GECCO 2001*. Morgan Kaufmann, 2001. nlin.AO/0105029.
- [Ste03] STEPHENS, CHRISTOPHER R.: *The renormalization group and the dynamics of genetic systems*. To be published in *Acta Physica Slovaca*, cond-mat/0210271, 2003.
- [SW99] STEPHENS, CHRISTOPHER R. and HENRI WAELBROECK: *Schemata evolution and building blocks*. *Evolutionary Computation*, 7(2):109–124, 1999. nlin.AO/0006048.
- [Tou04] TOUSSAINT, MARC: *Notes on information geometry and evolutionary processes*. nlin.AO/0408040, 2004.
- [Vos99] VOSE, MICHAEL D.: *The simple genetic algorithm: foundations and theory*. MIT Press, Cambridge Massachusetts, 1999.
- [WL92] WEGMANN, HELMUT und JÜRGEN LEHN: *Einführung in die Statistik*. Teubner, Stuttgart, 1992.
- [Wri99] WRIGHT, ALDEN H.: *The Exact Schema Theorem*, 1999. Technical report, University of Montana, Missoula, MT 59812, USA, 1999. <http://www.cs.umt.edu/u/wright/papers/schema.pdf>.

Literaturverzeichnis

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Leipzig, den 14. Oktober 2004

Susanne Schindler