

Hausarbeit im Rahmen des Seminars

Evolutionäre Algorithmen

im Fachgebiet Entscheidungslehre
am Lehrstuhl für Quantitative Methoden der
Westfälischen Wilhelms-Universität Münster

Mutation und Rekombination: Dynamische Analysen

Themensteller: Prof. Dr. Ulrich Müller-Funk

vorgelegt von: Dirk Loss
Körnerstraße 7
48151 Münster
dloss@uni-muenster.de

vorgelegt am: 23.11.2001

Inhalt

1. Einleitung	1
2. Grundlagen	2
3. Analyse der Mutation	5
3.1. Art des Equilibriums	5
3.2. Transiente Analyse: Markov-Modellierung der Mutation	7
3.3. Annäherungsgeschwindigkeit an das Equilibrium	9
4. Analyse der Rekombination	12
4.1. Art des Equilibriums	12
4.2. Annäherungsgeschwindigkeit an das Equilibrium	13
4.2.1. Spezialfall: P_0 -Uniform-Rekombination	16
4.2.2. Spezialfall: Hyperebenen der Ordnung 2	19
5. Analyse der Kombination von Mutation und Rekombination	21
5.1. Art des Equilibriums	21
5.2. Annäherungsgeschwindigkeit an das Equilibrium	21
6. Zusammenfassung und Ausblick	22
A. Nebenrechnungen	23
A.1. Berechnung der Annäherung an das Uniform Equilibrium	23
A.2. Bestimmung des globalen Maximums für die Disruptionswahrscheinlichkeit	24
A.3. Bestimmung des globalen Maximums für die Konstruktionswahrscheinlichkeit	25

1. Einleitung

Evolutionäre Algorithmen sind heuristische Problemlösungsverfahren, die sich in ihrer Vorgehensweise an die Prinzipien der Evolution anlehnen und sich auf Computern implementieren lassen. Sie unterwerfen (im Anfang meist zufällig generierte) Populationen von Lösungsalternativen den aus der Evolutionstheorie bekannten Operatoren Selektion, Rekombination und Mutation. (vgl. Spears 1998, S. 4).

Evolutionäre Algorithmen werden in vielen unterschiedlichen Bereichen angewandt (vgl. Nissen 1994), am verbreitetsten ist der Einsatz bei Such- und Optimierungsproblemen. Mit der mathematischen Modellierung von Evolutionären Algorithmen und der Entwicklung entsprechender Theorien wird das Ziel verfolgt, für ein gegebenes Problem den geeigneten Algorithmus angeben zu können, inklusive optimaler Parametereinstellungen und Voraussagen über die Laufzeit sowie über die Qualität der Lösung (vgl. Mühlenbein und Mahnig 2000).

Dazu ist zunächst ein genaues Verständnis der Eigenschaften von Evolutionären Algorithmen nötig. Besonders interessant sind die Eigenschaften der verschiedenen genetischen Operatoren, und zwar sowohl einzeln als auch in Kombination miteinander.

Diese Arbeit untersucht die Eigenschaften der beiden genetischen Operatoren Mutation und Rekombination. Es handelt sich dabei um eine dynamische Analyse: Wie entwickelt sich eine Population *im Zeitablauf*, wenn sie einem der beiden Operatoren unterworfen wird? Nähert sie sich einem Gleichgewichtszustand an? Wie sieht das Gleichgewicht aus? Wie schnell geschieht die Annäherung und von welchen Parametern hängt die Annäherungsgeschwindigkeit ab? Es zeigt sich, dass dabei die Ergebnisse statischer Analysen eingehen, so dass auf diesen Untersuchungen aufgebaut werden kann.

Die Arbeit folgt im Wesentlichen Spears (Spears 1998, Kapitel 9) und analysiert Mutation und Rekombination aus dem Blickwinkel Genetischer Algorithmen. Bei Genetischen Algorithmen handelt es sich um eine der drei Hauptvarianten Evolutionärer Algorithmen neben dem Evolutionären Programmieren (Fogel) und den Evolutionsstrategien (Rechenberg). Die für diese Arbeit relevante Terminologie und Modellierung Genetischer Algorithmen werden daher in Kapitel 2 kurz dargestellt.

Im Hauptteil wird zunächst jeder Operator isoliert betrachtet: Kapitel 3 untersucht Mutation, Kapitel 4 die Rekombination. Im Anschluss werden die Ergebnisse auf den Fall "Mutation und Rekombination" übertragen.

Einige Nebenrechnungen sind im Anhang der Arbeit zu finden.

2. Grundlagen

Ein Genetischer Algorithmus arbeitet auf einer *Population* von P *Individuen*, die jeweils die gleiche Anzahl L *Gene* besitzen. Individuen werden als *Strings* der Länge L dargestellt; an jeder Position des Strings denkt man sich ein Gen.¹ Gene treten in C unterschiedlichen Ausprägungen auf, die als *Allele* bezeichnet werden. Insgesamt sind auf diese Weise C^L verschiedene Strings möglich, die sich als Wörter der Länge L über einem Alphabet \mathcal{A} betrachten lassen (mit $\text{card}(\mathcal{A}) = C$).²

Beispiel: Im String AABDA treten die Allele A (an den Positionen 1, 2 und 5), B (an der Position 3) und D (an der Position 4) auf. Aus den drei Allelen A, B und D lassen sich $3^5 = 243$ verschiedene Strings der Länge 5 bilden.

Unter einer *Hyperebene* H versteht man ein String, der an einigen Stellen das Wildcard-Symbol $\#$ enthält.³ Wildcard-Symbole sind als “don’t care”-Felder oder Joker zu verstehen. Die Positionen, an denen kein Wildcard-Symbol auftritt, werden als *definierende Positionen* der Hyperebene bezeichnet. Die *Ordnung* k einer Hyperebene ist dabei die Zahl definierender Positionen; eine Hyperebene der Ordnung k wird notiert als H_k . Die *definierende Länge* L_1 einer Hyperebene ist der Abstand zwischen den beiden äußersten definierenden Positionen.

Beispiel: Die Hyperebene $H_2 = A\#\#D\#$ hat die Ordnung 2 (zwei Nicht-Wildcard-Symbole A und D) und die definierende Länge 3 (Abstand zwischen A und D).

Der allgemeine Ablauf eines Genetischen Algorithmus ist in Tabelle 2.1 auf Seite 3 dargestellt. Nach der initialen Bildung einer Population werden fortlaufend immer wieder die Schritte 4 bis 12 durchgeführt. Jeder Durchlauf dieser äußeren Schleife erzeugt aus der vorhandenen Population eine neue Population. Nach dem Ersetzen der alten Population durch die neue wird die Variable t um eins erhöht, d.h. die logische Zeit schreitet mit jeder neuen Generation um eine Einheit fort. In diesem Sinne ist der Genetische Algorithmus ein zeitdiskreter Prozess, in dem zu jedem Zeitpunkt eine neue Generation von Individuen erzeugt wird, welche die alte ersetzt.

In dieser Arbeit werden die Operationen der Schritte 8 und 9 näher untersucht, die Rekombination von zwei Strings zu zwei Nachkommen sowie die Mutation von Strings.

Ablauf der Rekombination

Die Rekombination ist der Hauptoperator bei Genetischen Algorithmen. Rekombination erzeugt aus zwei Strings S_i und S_j zwei neue Strings. Dabei können unterschiedliche Verfahren angewendet werden. Die gängigsten Verfahren sind:

¹Positionen werden in dieser Arbeit von 1 an durchnummeriert.

²Traditionell arbeiten die meisten Genetischen Algorithmen auf binären Strings, d.h. $\mathcal{A} = \{0, 1\}$ und $C = 2$. In dieser Arbeit wird der allgemeinere Fall $C \geq 2$ betrachtet. Um dies auch in den Beispielen deutlich zu machen, werden die Allele dort durchgehend als Großbuchstaben dargestellt.

³Eine Veranschaulichung des Begriffs Hyperebene findet sich z.B. bei Whitley (Whitley 1994)

```

1  Generiere Ausgangspopulation
2  t = 0
3  Wiederhole bis Abbruchbedingung erfüllt ist:
4      Bewerte Fitness jedes Strings
5      Neue Population = {}
6      Wiederhole bis neue Population vollständig ist:
7          Selektiere zwei Strings als Eltern
8          Rekombiniere die Strings, d.h. erzeuge zwei Nachkommen
9          Mutiere die Nachkommen
10         Füge Nachkommen der neuen Population hinzu
11         Ersetze alte Population durch neue Population
12         t = t+1
13  Gib Ergebnisse aus

```

Tabelle 2.1.: Genetischer Algorithmus im Pseudo-Code (vgl. Nissen 1994, S. 27)

- 1-point-Rekombination⁴

Es wird gleichverteilt zufällig und für beide Strings identisch ein Schnittpunkt (Cut-Point oder Crossover-Point) bestimmt. An allen Positionen hinter dem Schnittpunkt werden die Allele zwischen den beiden Strings ausgetauscht.

Beispiel: (auszutauschende Allele werden sind fett gedruckt)

$$\begin{array}{l}
 S_i : AB|DC \rightarrow ABAB \\
 S_j : BB|AB \rightarrow BBDC
 \end{array}$$

Bei Strings der Länge L gibt es L mögliche Schnittpunkte: fällt der Schittpunkt vor die Position 1, werden alle Allele ausgetauscht, fällt er im anderen Extremfall vor die Position L , wird nur das letzte Allel ausgetauscht.⁵

- n -point-Rekombination

Dieses Verfahren ist die Verallgemeinerung der 1-Point-Rekombination: Hier werden n verschiedene Schnittpunkte “ohne Zurücklegen” gleichverteilt zufällig ausgewählt. Jeder zweite Bereich zwischen zwei Schnittpunkten wird ausgetauscht.

Beispiel ($n = 3$):

$$\begin{array}{l}
 S_i : AB|DC|AAB|B \rightarrow ABABAABC \\
 S_j : BB|AB|DDD|C \rightarrow BBDCDDDB
 \end{array}$$

⁴Die Operatoren werden auch als 1-point-, n -point- und Uniform-Crossover bezeichnet. Die Begriffe sind gleichwertig (vgl. Pohlheim 1999, S. 14).

⁵Der $L + 1$ -te Fall, dass der Schnittpunkt *hinter* die letzte Position fallen könnte, wird ausgenommen, da dann überhaupt keine Rekombination stattfinden würde.

- P_0 -Uniform-Rekombination

Es wird nacheinander für jede Position ein Bernoulli-Experiment durchgeführt: Mit der Wahrscheinlichkeit P_0 werden die Allele an der betreffenden Position ausgetauscht, andernfalls bleiben sie unverändert. An welchen Positionen Allele ausgetauscht werden, lässt sich in Form einer Bitmaske notieren, in der 1 einen Austausch und 0 keinen Austausch der Allele bedeutet.

Beispiel: (Bitmaske = 00110001)

$$\begin{array}{l} S_i : \text{ABDCAABB} \quad \text{ABABAABC} \\ S_j : \text{BBABDDDC} \quad \rightarrow \quad \text{BBDCCDDDB} \end{array}$$

Ablauf der Mutation

Die beiden durch die Rekombination der Eltern entstandenen Strings werden einer Mutation unterzogen. Dies geschieht für jeden String einzeln, und zwar auf folgende Weise: Für jedes Gen des Strings wird in einem Bernoulli-Experiment entschieden, ob es mutiert wird oder nicht. Mit der Wahrscheinlichkeit μ wird das Gen mutiert, d.h. das an dieser Stelle zu findende Allel $\alpha \in \mathcal{A}$ wird durch eines der anderen Allele $\bar{\alpha} \in \mathcal{A} \setminus \{\alpha\}$ ersetzt. Zu welchem der $C - 1$ anderen Allele die Mutation erfolgt, wird durch ein Laplace-Experiment entschieden, d.h. alle $C - 1$ anderen Allele kommen mit der gleichen Wahrscheinlichkeit in Frage.⁶

⁶Im Fall binärer Strings ($C = 2$) bedeutet eine Mutation immer das Wechseln des Bits von 0 auf 1 oder umgekehrt. Da das Ergebnis der Mutation feststeht, ist in diesem Fall kein Laplace-Experiment nötig.

3. Analyse der Mutation

In diesem Kapitel wird der genetische Operator “Mutation” isoliert betrachtet, d.h. es werden keine Rekombinationen durchgeführt. Auch der Selektionsoperator bleibt unberücksichtigt; nacheinander wird jedes Individuum dem Operator “Mutation” unterworfen. Der so veränderte Ablauf des Genetischen Algorithmus ist in Tabelle 3.1 dargestellt.

1	Generiere Ausgangspopulation
2	$t = 0$
3	Wiederhole bis Abbruchbedingung erfüllt ist:
4	Neue Population = {}
5	Für alle Individuen in der Population:
6	Mutiere das Individuum
7	Füge das mutierte Individuum der neuen Population hinzu
8	Ersetze alte Population durch neue Population
9	$t = t+1$
10	Gib Ergebnisse aus

Tabelle 3.1.: Genetischer Algorithmus bei ausschließlicher Mutation

In den nächsten Abschnitten werden folgende Aspekte näher untersucht:

- Was passiert mit der Population, wenn sehr oft mutiert wird?
- Gegeben sei eine Population; welcher Zustand lässt sich nach einer bestimmten Anzahl von Mutationen erwarten?
- Wie schnell verändert sich die Population und wovon hängt diese Geschwindigkeit ab?

3.1. Art des Equilibriums

Im Mutationsschritt wird nacheinander für jede Position eines Strings, d.h. für jedes Gen, entschieden, ob es verändert wird. Wird die Mutation durchgeführt, wird das an dieser Stelle vorhandene Allel α durch irgendein anderes Allel ersetzt. Alle Allele des Alphabets sind als Ergebnis möglich.¹ Insgesamt bedeutet dies, dass durch Mutation alle theoretisch denkbaren Strings auch erzeugt werden können.

¹Mit Ausnahme von α , sonst handelt es sich nicht um eine wirkliche Veränderung. Von irgendeinem anderen Allel $\alpha' \in \bar{\alpha}$ kann im nächsten Schritt jedoch auch wieder das Allel α durch Mutation erreicht werden.

Setzt man eine Population ausschließlich der Mutation aus, so nähert sie sich einem Gleichgewichtszustand an, in dem alle möglichen Strings gleich wahrscheinlich sind (Spears 1998, S. 107):

$$\lim_{t \rightarrow \infty} p_S^{(t)} = \frac{1}{C^L} = \prod_{\ell=1}^L \frac{1}{C} \quad (3.1)$$

Dabei ist $p_S^{(t)}$ der erwartete Anteil eines beliebigen Strings S (aus den insgesamt C^L möglichen) an der Population zum Zeitpunkt t .

In diesem Gleichgewichtszustand, dem **Uniform Equilibrium**, sind die einzelnen Gene unabhängig voneinander. Auf die Art des Gleichgewichts haben weder die Mutationsrate μ noch die Anfangspopulation P_0 Einfluss, sondern nur die durch C und L bestimmte Anzahl der möglichen Strings. Ein Beispiel ist in Abbildung 3.2 auf Seite 6 dargestellt.

Eine Ausgangspopulation mit den fünf Individuen AAA, AAA, BAA, BAB und BBB werde fortlaufend der Mutation unterworfen: $P = 5$, $L = 3$, $C = 2$, $\mathcal{A} = \{A, B\}$. Die Anteile der beiden Allele A und B betragen: $p_A^{(0)} = \frac{9}{15} = 0.6$ und $p_B^{(0)} = \frac{6}{15} = 0.4$. Insgesamt sind $C^L = 2^3 = 8$ Allelkombinationen bzw. Strings möglich.

Die Anteile der Strings an der Population zum Zeitpunkt $t = 0$ sowie der jeweils erwartete Anteil für $t \rightarrow \infty$ können der folgenden Tabelle entnommen werden:

String	$p_{S_i}^{(0)}$	$\lim_{t \rightarrow \infty} p_{S_i}^{(t)}$
$S_1 = AAA$	$\frac{2}{5} = 0.4$	$0.5 \cdot 0.5 \cdot 0.5 = 0.125$
$S_2 = AAB$	0	$0.5 \cdot 0.5 \cdot 0.5 = 0.125$
$S_3 = ABA$	0	$0.5 \cdot 0.5 \cdot 0.5 = 0.125$
$S_4 = ABB$	0	$0.5 \cdot 0.5 \cdot 0.5 = 0.125$
$S_5 = BAA$	$\frac{1}{5} = 0.2$	$0.5 \cdot 0.5 \cdot 0.5 = 0.125$
$S_6 = BAB$	$\frac{1}{5} = 0.2$	$0.5 \cdot 0.5 \cdot 0.5 = 0.125$
$S_7 = BBA$	0	$0.5 \cdot 0.5 \cdot 0.5 = 0.125$
$S_8 = BBB$	$\frac{1}{5} = 0.2$	$0.5 \cdot 0.5 \cdot 0.5 = 0.125$

Mit der Zeit werden alle 8 möglichen Strings zu gleichen Anteilen in der Population vorkommen, unabhängig von ihren Anteilen an der Anfangspopulation. Wird dann ein Individuum herausgegriffen und eine Position darauf untersucht, welches Allel auftritt, treten alle Allele (hier: A und B) mit der gleichen Wahrscheinlichkeit auf.

Tabelle 3.2.: Beispiel zum Uniform Equilibrium

Da im Gleichgewichtszustand die Gene unabhängig voneinander sind, kann man jede Position einzeln betrachten und zählen, wie häufig die Allele dort auftreten. Dann bleibt aus dem Produkt $\prod_{\ell=1}^L \frac{1}{C}$ nur ein Faktor übrig – die Wahrscheinlich-

keit, an einer festen Position ℓ ein bestimmtes Allel α von den insgesamt C Allelen anzutreffen, beträgt im Uniform Equilibrium $1/C$.

Wie hoch ist dann die Anzahl X aller Individuen, die an dem festgelegten Locus das gewünschte Allel α haben? Da jedes Individuum unabhängig von den anderen mutiert wird, kann man sich ein P -faches Binomialexperiment mit der Erfolgswahrscheinlichkeit $1/C$ vorstellen:

$$\lim_{t \rightarrow \infty} P(X^{(t)} = i) = \binom{P}{i} \left(\frac{1}{C}\right)^i \left(1 - \frac{1}{C}\right)^{P-i} \quad (3.2)$$

3.2. Transiente Analyse: Markov-Modellierung der Mutation

Bisher wurde die Entwicklung der Population asymptotisch betrachtet, d.h. für $t \rightarrow \infty$. Da reale Algorithmen aber nicht beliebig lange laufen gelassen werden können, interessiert auch das transiente Verhalten. Deshalb wird im Folgenden untersucht, wie sich die Zahl der Gene mit einer bestimmten Ausprägung α mit der Zeit verändert (vgl. Spears 1998, S. 107ff). Die nicht betrachteten Allele werden gemeinsam mit $\bar{\alpha}$ bezeichnet. Da alle Positionen unabhängig voneinander mutiert werden, reicht es aus, nur eine feste Position ℓ zu betrachten.²

Die Population befinde sich Zeitpunkt t genau dann im Zustand $S_t = i$, wenn das Allel α zu diesem Zeitpunkt i -mal in der Population vorkommt (also in i Strings an der betrachteten Stelle ℓ). Bei insgesamt P Strings gibt es somit $P + 1$ mögliche Zustände: $S_t \in \{0, 1, \dots, P\}$.

Der Übergang von einem Zustand i in einen Nachfolgezustand j wird bestimmt durch die Verluste und Zugewinne an Allelen α :

$$j = i - \text{Verlust} + \text{Zugewinn}$$

Verluste ergeben sich, wenn α 's zu $\bar{\alpha}$'s mutiert werden. Umgekehrt treten Zugewinne auf, wenn $\bar{\alpha}$'s zu α 's mutiert werden:

$$j = i - \#[\alpha \rightarrow \bar{\alpha}] + \#[\bar{\alpha} \rightarrow \alpha] \quad (3.3)$$

Der Ausdruck $\#[\alpha \rightarrow \bar{\alpha}]$ bezeichnet hierbei die Anzahl der α 's, die zu $\bar{\alpha}$'s werden; $\#[\bar{\alpha} \rightarrow \alpha]$ beschreibt den umgekehrten Fall.

Da der Zustand zum Zeitpunkt $t + 1$ ausschließlich vom Vorgängerzustand zum Zeitpunkt t abhängt und sich der Mutationsmechanismus nicht ändert, lässt sich das Verhalten der Population mit einer homogenen Markov-Kette modellieren. Die Übergangswahrscheinlichkeit $p_{i,j} = P(S_t = j | S_{t-1} = i)$ bezeichne die Wahrscheinlichkeit, vom Zustand i in den Zustand j zu wechseln.

Bei der Berechnung der Übergangswahrscheinlichkeiten werden zwei Fälle unterschieden:

²Somit wird die Population in zwei Teile geteilt: zum einen diejenigen Individuen, die an der Position ℓ das Allel α besitzen und zum anderen die restlichen Individuen, die an der Position ℓ ein Nicht- α -Allel $\bar{\alpha}$ besitzen.

Fall 1: $j \geq i$ (Anzahl der α 's erhöht sich oder bleibt gleich)

Sei x die Anzahl der α 's, die zu $\bar{\alpha}$'s mutiert werden ($\#[\alpha \rightarrow \bar{\alpha}]$). Die Mutation von x Allelen geschieht mit der Wahrscheinlichkeit μ^x . Für die Auswahl der x Allele aus den i α 's gibt es $\binom{i}{x}$ Möglichkeiten. Die restlichen $i - x$ α 's werden *nicht* mutiert; dafür beträgt die Wahrscheinlichkeit $(1 - \mu)^{i-x}$.

Andererseits gibt es $P - i$ $\bar{\alpha}$'s in der Population. Wieviele von ihnen beim Übergang zum Zustand j zu α 's mutiert werden müssen, lässt sich berechnen, indem man x in Gleichung 3.3 einsetzt und nach $\#[\bar{\alpha} \rightarrow \alpha]$ auflöst:

$$\#[\bar{\alpha} \rightarrow \alpha] = x + j - i$$

An dieser Stelle wird deutlich, warum zwei Fälle unterschieden werden müssen: Damit $x + j - i$ auch für $x = 0$ (keine Mutationen von α zu $\bar{\alpha}$) nicht negativ werden kann, muss $j \geq i$ gelten.

Wenn ein Allel $\bar{\alpha}$ mutiert wird, ist das Ergebnis durchschnittlich nur in einem von $C - 1$ Fällen das Allel α . Damit ergibt sich für jedes einzelne Allel $\bar{\alpha}$ eine Wahrscheinlichkeit, zum Allel α zu mutieren, von $\mu/(C - 1)$. Die nötigen $x + j - i$ solcher Mutationen treten mit der Wahrscheinlichkeit $(\mu/(C - 1))^{x+j-i}$ auf. Für die Auswahl dieser $x + j - i$ Allele aus den $P - i$ $\bar{\alpha}$'s gibt es $\binom{P-i}{x+j-i}$ Möglichkeiten.

Aber nicht alle $\bar{\alpha}$'s werden mutiert. Wenn es $P - i$ $\bar{\alpha}$'s gibt und $x + j - i$ zu α 's mutiert werden, ist festgelegt, dass die restlichen $(P - i) - (x + j - i) = P - j - x$ $\bar{\alpha}$'s *nicht* zu α 's mutiert werden. Dies geschieht mit der Wahrscheinlichkeit $(1 - \mu/(C - 1))^{P-x-j}$.

Multipliziert man alle diese Wahrscheinlichkeiten und Binomialkoeffizienten, erhält man die Wahrscheinlichkeit, dass die Anzahl der α 's von i auf j steigt (bzw. gleich bleibt), wenn genau x α 's mutiert werden. Nun muss über alle zulässigen Werte von x summiert werden: Minimal werden überhaupt keine α 's mutiert ($x = 0$). Das Maximum für x ist durch zwei Bedingungen beschränkt. Erstens können nicht mehr als i α 's mutiert werden ($x \leq i$), zweitens können nicht mehr als $P - i$ $\bar{\alpha}$'s mutiert werden ($x + j - i \leq P - i \Leftrightarrow x \leq P - j$). Insgesamt ergibt sich somit für die Übergangswahrscheinlichkeit von Zustand i nach Zustand j :

$$p_{i,j} = \sum_{x=0}^{\min\{i, P-j\}} \binom{i}{x} \binom{P-i}{x+j-i} \mu^x \left(\frac{\mu}{C-1}\right)^{x+j-i} (1-\mu)^{i-x} \left(1 - \frac{\mu}{C-1}\right)^{P-j-x} \quad (3.4)$$

Fall 2: $j \leq i$ (Anzahl der α 's verringert sich oder bleibt gleich)

In diesem Fall bezeichne x die Anzahl der $\bar{\alpha}$'s, die zu α mutiert werden, x steht hier also für den Zugewinn. Die Wahrscheinlichkeit für x solcher Mutationen mit dem Ergebnis α beträgt $(\mu/(C - 1))^x$.

Die restlichen $P - i - x$ $\bar{\alpha}$'s werden nicht mutiert (Wahrscheinlichkeit: $(1 - \mu/(C - 1))^{P-i-x}$). Aus

$$j = i - \#[\alpha \rightarrow \bar{\alpha}] + x$$

folgt, dass dann $x+i-j$ α 's zu $\bar{\alpha}$'s mutiert werden müssen, um genau j α 's zu erhalten (Wahrscheinlichkeit: μ^{x+i-j}). Die restlichen der i α 's, also $i - (x + i - j) = j - x$ α 's werden *nicht* mutiert (Wahrscheinlichkeit: $(1 - \mu)^{j-x}$). Bezieht man die Anzahl der

Möglichkeiten mit ein, x $\bar{\alpha}$'s aus $P - i$ $\bar{\alpha}$'s und $x + i - j$ α 's aus i α 's auszuwählen, kommt man auf folgende Übergangswahrscheinlichkeit:

$$p_{i,j} = \sum_{x=0}^{\min\{P-i,j\}} \binom{i}{x+i-j} \binom{P-i}{x} \mu^{x+i-j} \left(\frac{\mu}{C-1}\right)^x (1-\mu)^{j-x} \left(1 - \frac{\mu}{C-1}\right)^{P-i-x} \quad (3.5)$$

Der Wertebereich für x wird dabei einerseits durch die Anzahl von $\bar{\alpha}$'s ($P - x$) beschränkt, andererseits können nicht mehr als i α 's mutiert werden ($x + i - j \leq i \Leftrightarrow x \leq j$).

Wenn sich die Anzahl der Allele α nicht verändert, d.h. für $i = j$, überschneiden sich die beiden Fälle und die Gleichungen liefern identische Übergangswahrscheinlichkeiten $p_{i,i}$. Im allgemeinen gilt jedoch $p_{i,i} \neq p_{j,i}$, da die Population dazu neigt, sich langfristig dem Uniform Equilibrium anzunähern. Die Gleichungen sind daher nicht symmetrisch. (vgl. Spears 1998, S. 109)

Die $P+1 \times P+1$ Übergangsmatrix $Q = [p_{i,j}]$ der Markov-Kette ist damit bestimmt. Für $0 < \mu < 1$ sind alle Einträge der Übergangsmatrix echt positiv.³ Damit ist die Übergangsmatrix regulär und die Markov-Kette ergodisch, d.h. sie besitzt einen Steady-State.

Sei $p_i^{(t)} \equiv P(S_t = i)$ die Wahrscheinlichkeit, dass sich die Markov-Kette zum Zeitpunkt t im Zustand i befindet, und $p^{(t)} = (p_0^{(t)}, p_1^{(t)}, \dots, p_P^{(t)})^T$ der aus diesen Wahrscheinlichkeiten zusammengesetzte stochastische Vektor. Dann gilt:

$$p^{(t)} = p^{(0)} Q^t \quad (3.6)$$

Dabei bezeichnet $p^{(0)}$ den Vektor der Wahrscheinlichkeiten für die einzelnen Zustände im Anfangszeitpunkt und Q^t die t -mal mit sich selbst multiplizierte Übergangsmatrix. Mit dieser Gleichung lassen sich die erwarteten Anzahlen eines Allels für beliebige Zeitpunkte bestimmen, sofern man die Anfangsbedingungen kennt.

Durch eine Eigenwertbetrachtung ließe sich nun prinzipiell der Steady-State der Markov-Kette ermitteln. Aufgrund der Kompliziertheit der Übergangsmatrix ist dies von Spears nur numerisch nachvollzogen worden. In seinen Versuchen ergab sich als Steady-State-Verteilung immer die Binomialverteilung.

$$\lim_{t \rightarrow \infty} p_i^{(t)} = \binom{P}{i} \left(\frac{1}{C}\right)^i \left(1 - \frac{1}{C}\right)^{P-i} \quad (3.7)$$

3.3. Annäherungsgeschwindigkeit an das Equilibrium

Nach Gleichung 3.1 hängen die Anteile der Strings im Uniform Equilibrium nur von der Anzahl der möglichen verschiedenen Strings ab, also von C und L . Unabhängig davon, welche Strings in der Anfangspopulation auftreten und wie häufig mutiert wird, führt eine fortlaufende Anwendung des Mutationsoperators die Population asymptotisch betrachtet immer in das Uniform Equilibrium.

³Für $\mu = 0$ wird niemals mutiert und die Population bleibt für immer im Anfangszustand. Den Fall $\mu = 1$ betrachtet Spears nicht weiter.

Die Anfangspopulation und die Mutationsrate haben jedoch sehr wohl Einfluss auf *Geschwindigkeit*, mit der das Equilibrium erreicht wird. Dies wird im Folgenden näher untersucht.

Sei $X_\alpha^{(t)}$ die Anzahl der Gene mit der Ausprägung α (kurz: “die Anzahl des Allels α ”) in der Population zum Zeitpunkt t . Die erwartete Anzahl von Allelen α in der Nachfolgeneration hängt davon ab, wie viele Allele im gegenwärtigen Mutations-schritt verloren gehen bzw. dazukommen:

$$\mathbb{E}(X_\alpha^{(t+1)}) = \mathbb{E}(X_\alpha^{(t)}) - \mathbb{E}(\text{Verlust}) + \mathbb{E}(\text{Gewinn})$$

Verluste entstehen, wenn α 's mutiert werden.

$$\mathbb{E}(\text{Verlust}) = \mathbb{E}(X_\alpha^{(t)}) \cdot \mu$$

Zugewinne treten auf, wenn Nicht- α -Allele mutiert werden und das Ergebnis der Mutation ein α -Allel ist. Insgesamt gibt es $P \cdot L$ Allele, also gibt es $P \cdot L - \mathbb{E}(X_\alpha^{(t)})$ Nicht- α -Allele. Wenn eine Mutation erfolgt, sind $C - 1$ Ergebnisse möglich, von denen nur eines das Allel α ist (vgl. Seite 8).

$$\mathbb{E}(\text{Gewinn}) = \mu \frac{1}{C-1} (P \cdot L - \mathbb{E}(X_\alpha^{(t)}))$$

Damit ergibt sich:

$$\mathbb{E}(X_\alpha^{(t+1)}) = \mathbb{E}(X_\alpha^{(t)}) - \mathbb{E}(X_\alpha^{(t)}) \cdot \mu + \frac{\mu}{C-1} (P \cdot L - \mathbb{E}(X_\alpha^{(t)}))$$

Interessiert man sich nur für den erwarteten *Anteil* p_α des Allels α an der Population, dividiert man die Gleichung durch die Gesamtanzahl der Allele in der Population ($P \cdot L$) und erhält

$$p_\alpha^{(t+1)} = p_\alpha^{(t)} - \mu p_\alpha^{(t)} + \frac{\mu}{C-1} (1 - p_\alpha^{(t)}) \quad (3.8)$$

Zur besseren Handhabbarkeit nähert man die zeitdiskrete Funktion $p_\alpha^{(t)}$ mit $t \in N_0$ an durch eine zeitkontinuierliche Funktion $p_\alpha^{(t)}$ mit $t \in \mathbb{R}, t \geq 0$.

Näherungsweise gilt:⁴

$$\frac{dp_\alpha^{(t)}}{dt} \approx \frac{p_\alpha^{(t+\Delta t)} - p_\alpha^{(t)}}{\Delta t}$$

Mit jeder Generation wird die logische Zeit um eine Einheit erhöht, d.h. $\Delta t = 1$:

$$\frac{dp_\alpha^{(t)}}{dt} \approx p_\alpha^{(t+1)} - p_\alpha^{(t)} = -\mu p_\alpha^{(t)} + \left(\frac{\mu}{C-1} \right) (1 - p_\alpha^{(t)})$$

Zu lösen ist also die folgende Differentialgleichung (Spears 1998, S. 111):

$$\frac{dp_\alpha^{(t)}}{dt} = \left(\frac{\mu}{C-1} \right) (1 - C p_\alpha^{(t)}) \quad (3.9)$$

⁴vgl. Definition des Differentialquotienten bzw. Taylorentwicklung von $\frac{dp}{dt}$ um t

Durch Anwendung des Verfahrens der Trennung der Variablen ergibt sich folgende Lösung (Spears 1998, S. 111):⁵

$$p_{\alpha}^{(t)} = \frac{1}{C} + \left(p_{\alpha}^{(0)} - \frac{1}{C} \right) e^{\frac{-C\mu t}{C-1}} \quad (3.10)$$

Die Geschwindigkeit, mit der sich die Population dem Uniform Equilibrium nähert, wird somit von drei Parametern beeinflusst: der Mutationsrate μ , den Anfangsbedingungen $p_{\alpha}^{(0)}$ und der Kardinalität C des Alphabets. Die Gleichung 3.10 besteht aus drei Teilen: Der erste Summand $\frac{1}{C}$ stellt den erwarteten Anteil des Allels α im Uniform Equilibrium dar. Der Klammerausdruck gibt die Abweichung des zum Anfangszeitpunkt vorhandenen Anteils vom Anteil im Equilibrium an. Diese Differenz nimmt mit steigendem t exponentiell ab.⁶

In Abbildung 3.1 ist in einem Beispiel für ein binäres Alphabet dargestellt, wie sich der erwartete Anteil eines Allels an der Population mit der Zeit dem Gleichgewicht annähert. Zu Anfang besitzen *alle* Individuen an der betrachteten Position das Allel α . Bei einer Mutationsrate von $\mu = 0.01$ ist nach mehreren hundert Generationen nur noch in 50 Prozent der Fälle das Allel α zu erwarten. In den anderen Fällen erwartet man das andere Allel $\bar{\alpha}$.

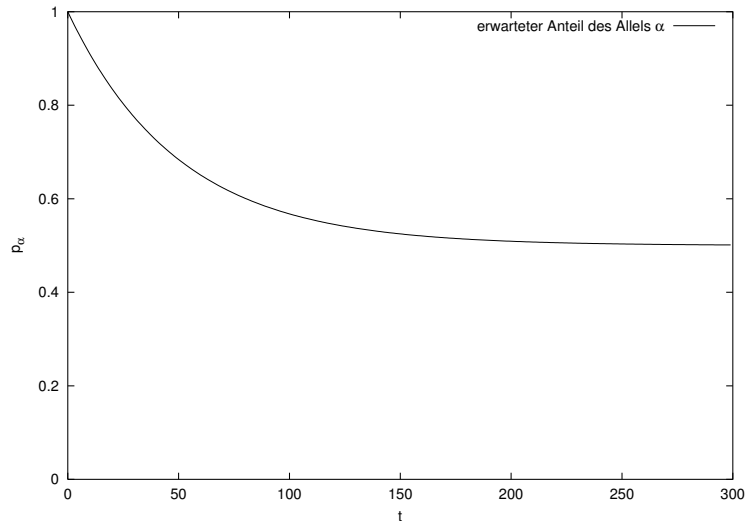


Abbildung 3.1.: Annäherung an das Uniform Equilibrium ($C = 2$, $\mu = 0.01$, $p_{\alpha}^{(0)} = 1$)

Auch anhand dieser Modellerierung lässt sich erkennen, dass nur C einen Einfluss auf den erreichten Endzustand hat:

$$\lim_{t \rightarrow \infty} p_{\alpha}^{(t)} = \frac{1}{C}$$

Im Uniform Equilibrium kann man erwarten, dass jedes der C verschiedenen Allele zu gleichen Anteilen auftritt.

⁵Herleitung siehe Anhang

⁶Für binäre Alphabete ($C = 2$) erfolgt die Annäherung am schnellsten: Dann gilt $-C/(C-1) = -2$; für höhere Werte von C nimmt die Geschwindigkeit ab: $\lim_{C \rightarrow \infty} -C/(C-1) = -1$.

4. Analyse der Rekombination

In diesem Kapitel soll eine Population ausschließlich rekombiniert werden. Der entsprechende Ablauf ist in Tabelle 4.1 dargestellt. Es werden zufällig Paare von Individuen gebildet, die jeweils zwei Nachkommen erzeugen. Jedes Individuum wird pro Generation genau einmal als Elternteil berücksichtigt.¹

```
1  Generiere Ausgangspopulation
2  t = 0
3  Wiederhole bis Abbruchbedingung erfüllt ist:
4      Neue Population = {}
5      Für alle noch nicht als Eltern selektierten Individuen:
6          Selektiere zufällig (gleichverteilt) zwei Eltern
7          Rekombiniere die Eltern, d.h. erzeuge Nachkommen
8          Füge Nachkommen der neuen Population hinzu
9      Ersetze alte Population durch neue Population
10     t = t+1
11  Gib Ergebnisse aus
```

Tabelle 4.1.: Genetischer Algorithmus bei ausschließlicher Rekombination

4.1. Art des Equilibriums

Bei der Rekombination werden zwischen zwei Strings an bestimmten Stellen die Allele ausgetauscht. Die in diesem Kapitel untersuchten Operatoren n -point-Rekombination und P_0 -Uniform-Rekombination können die *Positionen* der Allele allerdings nicht verändern:² Befand sich ein beliebiges Allel α vor der Rekombination an Position ℓ in einem String, so befindet es sich auch nach der Rekombination an der Position ℓ – allerdings ggf. in einem anderen String. Bezogen auf eine Position ℓ bleibt somit das Verhältnis der Allele stets konstant: Wieviel Prozent der Individuen zum Zeitpunkt

¹Sowohl Spears als auch Booker wollen die Selektion als Operator unberücksichtigt lassen, geben aber nicht an, auf welche Weise die Zusammenstellung der Strings zu Elternpaaren erfolgt. Das hier explizit geschilderte Verfahren versucht den Einfluss der Selektion zu minimieren, indem Strings nicht mehrmals als Elternteile zugelassen werden und die Auswahl gleichverteilt zufällig erfolgt.

²Es gibt andere Rekombinationsoperatoren, welche die Positionen von Allelen ändern können, z.B. die oft bei der Lösung des Travelling Salesman Problems verwendeten Operatoren “partially matched crossover” und “ordered crossover”. Diese werden hier nicht untersucht.

t an der Position ℓ das Allel α besitzen, werde durch $p_{\alpha_\ell}^{(t)}$ dargestellt. Es gilt dann: $p_{\alpha_\ell}^{(t)} = p_{\alpha_\ell}^{(0)} \forall t$.

Verfolgt man eine Population im Zeitablauf, so reduziert die Anwendung des Rekombinationsoperators mit der Zeit die Korrelation zwischen den Ausprägungen der Gene an unterschiedlichen Positionen eines Strings (vgl. Juels u. a. 1993, S. 9).

Sei S ein String mit den String S mit den L Allelen $(\alpha_1, \dots, \alpha_L)$ und bezeichne $p_S^{(t)}$ den erwarteten Anteil des Strings S an der Population zum Zeitpunkt t bzw. die Wahrscheinlichkeit, zum Zeitpunkt t bei einmaligem Ziehen aus der Population den String S anzutreffen. Dann gilt (Spears 1998, Seite 99):

$$\lim_{t \rightarrow \infty} p_{S_i}^{(t)} = \prod_{\ell=1}^L p_{\alpha_\ell}^{(0)} \quad (4.1)$$

Die Population nähert sich also bei ausschließlicher Rekombination mit der Zeit einem Zustand, in dem alle Gene eines Strings unabhängig voneinander sind. Dieser Gleichgewichtszustand wird als **Robbins' Equilibrium** bezeichnet.³ Den Beweis hat die Mathematikerin Hilda Geiringer schon 1944 erbracht. Geiringers Theorem gilt unabhängig von der Kardinalität des gewählten Alphabets. Ein Beispiel ist in Abbildung 4.2 auf Seite 14 dargestellt.

Zu beachten ist, dass es sich bei p_S um einen erwarteten und nicht um einen tatsächlich beobachtbaren Anteil handelt. In einer endlichen Population können die vorausgesagten Anteile der Strings gar nicht immer exakt erreicht werden (vgl. Mühlenbein und Mahnig 2000, S. 5). In Beispiel 1 mit einer Populationsgröße von $P = 5$ kann z.B. der Anteil des Strings BAA nicht exakt 0.288 betragen, sondern in einer real beobachteten Population bestenfalls $\frac{1}{5} = 0.2$ oder $\frac{2}{5} = 0.4$. Daher lässt sich nicht ohne weiteres beurteilen, ob sich eine beobachtete Population im Equilibrium befindet. Mühlenbein und Mahnig definieren dazu ein Abstandsmaß zwischen den beobachteten Anteilen und den theoretischen Anteilen im Equilibrium und untersuchen, wann der Abstand minimal wird (vgl. Mühlenbein und Mahnig 2000, S. 5). Spears geht in seiner Dissertation auf diese Frage nicht ein.

4.2. Annäherungsgeschwindigkeit an das Equilibrium

Um festzustellen, wie schnell sich eine Population an das Equilibrium annähert, muss man jeden möglichen String einzeln betrachten und berechnen, wie sich die Anteile des Strings an der Population im Zeitablauf verändern. Der Anteil kann zunehmen (wenn Strings dieses Typs in manchen Individuen konstruiert werden) und abnehmen (wenn Strings dieses Typs manche Rekombinationen nicht überleben und auf diese Weise verloren gehen). So kommt man wie in Kapitel 3.3 approximativ auf eine Differentialgleichung (Spears 1998, S. 100):

$$\frac{dp_{S_i}^{(t)}}{dt} = -\text{Verlust}_{S_i}^{(t)} + \text{Zugewinn}_{S_i}^{(t)}$$

³In der Populationsgenetik wird dieses Gleichgewicht auch als "linkage equilibrium" bezeichnet (vgl. Mühlenbein und Mahnig 2000, S. 4).

Eine Ausgangspopulation mit den fünf Individuen AAA, AAA, BAA, BAB und BBB werde fortlaufend einer Rekombination unterworfen ($P = 5$, $L = 3$, $C = 2$, $\mathcal{A} = \{A, B\}$). Bei zwei der fünf Individuen befindet sich an der Position 1 das Allel A ($p_{A_1}^{(0)} = \frac{2}{5} = 0.4$); bei drei Individuen befindet sich an Position 1 ein B ($p_{B_1}^{(0)} = \frac{3}{5} = 0.6$). Für die Positionen 2 und 3 ergeben sich die folgenden Anteile: $p_{A_2}^{(0)} = \frac{4}{5} = 0.8$, $p_{B_2}^{(0)} = \frac{1}{5} = 0.2$; $p_{A_3}^{(0)} = \frac{3}{5} = 0.6$, $p_{B_3}^{(0)} = \frac{2}{5} = 0.4$. Da beide Allele in der Ausgangspopulation an allen Positionen vorkommen, sind alle $C^L = 2^3 = 8$ denkbaren Allelkombinationen durch Rekombination zu erreichen und damit als Strings möglich.

Die Anteile der Strings an der Population zum Zeitpunkt $t = 0$ sowie der jeweils erwartete Anteil für $t \rightarrow \infty$ können der folgenden Tabelle entnommen werden:

String	$p_{S_i}^{(0)}$	$\lim_{t \rightarrow \infty} p_{S_i}^{(t)}$
$S_1 = AAA$	$\frac{2}{5} = 0.4$	$0.4 \cdot 0.8 \cdot 0.6 = 0.192$
$S_2 = AAB$	0	$0.4 \cdot 0.8 \cdot 0.4 = 0.128$
$S_3 = ABA$	0	$0.4 \cdot 0.2 \cdot 0.6 = 0.048$
$S_4 = ABB$	0	$0.4 \cdot 0.2 \cdot 0.4 = 0.032$
$S_5 = BAA$	$\frac{1}{5} = 0.2$	$0.6 \cdot 0.8 \cdot 0.6 = 0.288$
$S_6 = BAB$	$\frac{1}{5} = 0.2$	$0.6 \cdot 0.8 \cdot 0.4 = 0.192$
$S_7 = BBA$	0	$0.6 \cdot 0.2 \cdot 0.6 = 0.072$
$S_8 = BBB$	$\frac{1}{5} = 0.2$	$0.6 \cdot 0.2 \cdot 0.4 = 0.048$

Der anfängliche Anteil Strings der S_1 , S_6 und S_8 ist größer als im Equilibrium. Daher haben sie in der Summe mehr Verluste als Zugewinne zu erwarten. Die Anteile der Strings S_2 , S_3 , S_4 , S_5 und S_7 wird sich dagegen mit der Zeit erhöhen; die erwarteten Zugewinne sind größer als die erwarteten Verluste.

Tabelle 4.2.: Beispiel zu Robbins' Equilibrium

Dabei ist $p_{S_i}^{(t)}$ der Anteil des betrachteten Strings S_i zum Zeitpunkt t .

Verluste

Der String S_i kann bei einer Rekombination nur dann verloren gehen, wenn der String S_j , mit dem er rekombiniert wird, sich an mindestens zwei Positionen von S_i unterscheidet. Denn sind beide Strings gleich, bleiben sie bei der Rekombination unverändert, egal welche Positionen ausgetauscht werden.

$$\begin{aligned} S_i &: AA|\mathbf{AA} \rightarrow AAAA \\ S_j &: AA|\mathbf{AA} \rightarrow AAAA \end{aligned}$$

Unterscheiden Sie sich an genau einer Position, kann höchstens diese Position zwischen den Strings ausgetauscht werden; aus S_i wird dann S_j und aus S_j wird S_i , beide Strings bleiben aber erhalten:

$$\begin{aligned} S_i &: AA|\mathbf{AA} \rightarrow AABA \\ S_j &: AA|\mathbf{BA} \rightarrow AAAA \end{aligned}$$

Anders verhält es sich, wenn zwei oder mehr Positionen unterschiedlich sind. Zum Beispiel kann der String $S_i = AAAA$ bei der Rekombination mit $S_j = AABB$ verloren gehen – nämlich genau dann, wenn der Schnittpunkt zwischen die dritte und vierte Position fällt:

$$\begin{array}{l} S_i : AAA|\mathbf{A} \rightarrow AAAB \\ S_j : AAB|\mathbf{B} \rightarrow AABA \end{array}$$

Für die Frage, ob S_i die Rekombination überlebt, spielen nur die k Positionen eine Rolle, an denen sich S_i und S_j unterscheiden; denn wo die Allele der Eltern übereinstimmen, kann sich keine Veränderung ergeben. Belegt man die übereinstimmenden Positionen mit einem Wildcard-Symbol #, erhält man aus S_i eine Hyperebene oder Ordnung k . S_i überlebt genau dann die Rekombination, wenn die entsprechende Hyperebene über den sich unterscheidenden Positionen (im Beispiel die Hyperebene ##AA) die Rekombination überlebt. Somit kann man für die Berechnung der Verlustrate auf die Ergebnisse *statistischer* Analysen in bezug auf die Überlebenswahrscheinlichkeit von Hyperebenen bei der Rekombination zurückgreifen. Es gilt (Spears 1998, S. 100):

$$\text{Verlust}_{S_i}^{(t)} = \sum_{S_j} p_{S_i}^{(t)} \cdot p_{S_j}^{(t)} \cdot P_d(H_k) \quad \text{mit} \quad 2 \leq k \equiv \Delta\{S_i, S_j\} \leq L$$

Dabei ist $p_{S_i}^{(t)} \cdot p_{S_j}^{(t)}$ die Wahrscheinlichkeit, dass S_i mit einem bestimmten String S_j rekombiniert wird. Da jedes Individuum pro Zeiteinheit genau einmal als Elternteil in eine Rekombination eingeht (vgl. Seite 12), ist die Wahrscheinlichkeit für einen String, rekombiniert zu werden, gleich seinem Anteil an der Population zum jeweiligen Zeitpunkt.

$P_d(H_k)$ ist die Wahrscheinlichkeit, dass die Hyperebene über den k Stellen, an denen sich S_i von S_j unterscheidet, die Rekombination nicht überlebt. Sie wird im Folgenden als *Disruptionswahrscheinlichkeit* bezeichnet. Wie oben gezeigt, spielen dabei nur Werte für $k \geq 2$ eine Rolle und maximal können sich die Strings an allen (d.h. an L) Stellen unterscheiden.

Um den erwarteten Verlust auszurechnen, muss die Wahrscheinlichkeit für einen Verlust bei der Rekombination mit S_j noch mit der Anzahl der bei jedem Verlust zerstörten Strings S_i multipliziert werden. Da bei jedem Rekombinationsvorgang nur genau *ein* Exemplar des Strings S_i verloren gehen kann, ist der Faktor 1 und kann in der Gleichung weggelassen werden.⁴

Die sich ergebenden Verluste werden danach über alle möglichen Rekombinationspartner von S_i aufsummiert.

Zugewinn

Für den Zugewinn ergibt sich ein ganz ähnlicher Ansatz. Der Anteil eines Strings S_i an der Population kann sich nur dann erhöhen, wenn er bei der Rekombination von zwei Strings S_h und S_j neu konstruiert wird. Damit eine erfolgreiche Konstruktion möglich ist, müssen an jeder Position entweder S_h oder S_j (oder beide) mit dem Allel von S_i an dieser Position übereinstimmen (also das “richtige” Allel besitzen).

⁴Geht kein Exemplar von S_i verloren, ist dies kein Verlust. Wenn beide Eltern S_i sind, bleiben beide erhalten.

Positionen, an denen sowohl S_h als auch S_i mit S_i übereinstimmen, können außer Acht gelassen werden – die Rekombination liefert an diesen Stellen mit Sicherheit das gewünschte Ergebnis.

Interessant für die Analyse sind also wieder nur die k Stellen, an denen sich S_h und S_j unterscheiden. Echte Zugewinne treten nur für $k \geq 2$ auf: für $k = 1$ gibt es sowohl vorher als auch nachher genau einen String, der mit S_i identisch ist (vgl. Seite 14). So lässt sich die Konstruktion des Strings S_i als Konstruktion einer Hyperebene der Ordnung k darstellen; m Gene kommen dabei von S_h und $n = k - m$ Allele von S_j :

$$\begin{array}{l} S_h : A \overbrace{\text{AAAA}}^m \text{BBB A} \\ S_j : A \text{BBBB} \underbrace{\text{AAA A}}_n \end{array} \rightarrow S_i : \text{ABBB BBBBA} \quad \text{AAAAA AAAA}$$

So ergibt sich für den erwarteten Zugewinn:

$$\text{Zugewinn}_{S_i}^{(t)} = \sum_{S_h, S_j} p_{S_h}^{(t)} p_{S_j}^{(t)} P_c(H_k | H_m \wedge H_n) \quad \text{mit} \quad 2 \leq k \equiv \Delta(S_h, S_j) \leq L$$

Dabei sind $p_{S_h}^{(t)}$ und $p_{S_j}^{(t)}$ die Anteile der Strings S_h bzw. S_j zum Zeitpunkt t , $p_{S_h}^{(t)} p_{S_j}^{(t)}$ ist die Wahrscheinlichkeit, dass S_h mit S_j rekombiniert wird und $P_c(H_k | H_m \wedge H_n)$ ist die Wahrscheinlichkeit, dass bei der Rekombination von S_h mit S_j eine Hyperebene der Ordnung k und damit der String S_i konstruiert wird (im Weiteren als *Konstruktionswahrscheinlichkeit* bezeichnet).

Nach Einsetzen der richtigen Disruptions- und Konstruktionswahrscheinlichkeiten in die Verlust- bzw. Zugewinnterme erhält man so für jeden der C^L möglichen Strings eine Differentialgleichung. Da die Anteile der Strings voneinander abhängen, müssen alle Differentialgleichungen gleichzeitig erfüllt sein. Die Lösung dieses Systems von Differentialgleichungen würde die erwarteten Veränderungen der Population im Zeitablauf vorhersagen. Leider hat es sich aufgrund der im praktischen Anwendungsfall sehr großen Anzahl von Gleichungen (bei binären Strings der Länge 30 über eine Milliarde) und besonders wegen der komplizierten Berechnung der Disruptions- und Konstruktionswahrscheinlichkeiten als schwierig herausgestellt, eine allgemeine Lösung zu finden. Deshalb wird die Fragestellung in mehrfacher Hinsicht reduziert:

Zum einen versucht man nicht, die *absolute* Annäherungsgeschwindigkeit jedes Rekombinationsoperators zu berechnen, sondern interessiert sich nur für den *Vergleich* der unterschiedlichen Operatoren bzw. Parameter. Welcher von zwei gegebenen Rekombinationsoperatoren führt einen String bzw. eine Hyperebene schneller ins Equilibrium?

Zum anderen sind für die Beantwortung dieser Frage die Disruptions- bzw. Konstruktionswahrscheinlichkeiten für die Hyperebenen entscheidend. Diese sind in zwei Fällen besonders einfach zu berechnen: bei P_0 -Uniform-Rekombination und in bezug auf Hyperebenen der Ordnung 2. Diese beiden Fälle werden im folgenden näher untersucht.

4.2.1. Spezialfall: P_0 -Uniform-Rekombination

Wenn als Rekombinationsoperator P_0 -Uniform-Rekombination gewählt wird, lassen sich die Verlust- und Zugewinn-Terme der Differentialgleichungen folgendermaßen berechnen:

Verluste

Betrachtet man bei der Rekombination der Strings S_i und S_j nur die k Positionen an denen sich die Strings unterscheiden, kommt man zu zwei Hyperebenen $H_k^{(i)}$ und $H_k^{(j)}$ der Ordnung k . Die Hyperebene $H_k^{(i)}$ (und damit der String S_i) kann nur in zwei Fällen überleben: entweder, wenn alle k Allele ausgetauscht werden

$$\begin{array}{l} H_k^{(i)} : \mathbf{AAAA} \\ H_k^{(j)} : \mathbf{BBBB} \end{array} \rightarrow \begin{array}{l} \mathbf{BBBB} \\ \mathbf{AAAA} \end{array}$$

oder wenn gar kein Allel ausgetauscht wird:

$$\begin{array}{l} H_k^{(i)} : \mathbf{AAAA} \\ H_k^{(j)} : \mathbf{BBBB} \end{array} \rightarrow \begin{array}{l} \mathbf{AAAA} \\ \mathbf{BBBB} \end{array}$$

Für die Überlebenswahrscheinlichkeit einer Hyperebene der Ordnung k gilt hier daher:⁵

$$P_s(H_k, P_0) = P_0^k + (1 - P_0)^k$$

In allen anderen Fällen wird die Hyperebene bei der Rekombination zerstört:

$$P_d(H_k, P_0) = 1 - P_s(H_k, P_0) = 1 - P_0^k - (1 - P_0)^k \quad (4.2)$$

Diese Disruptionswahrscheinlichkeit wird für $P_0 = 0.5$ maximal.⁶ Da $P_d(H_k)$ in alle Summanden der Verlustterme eingeht, bedeutet dies, dass für $P_0 = 0.5$ die erwarteten Verluste pro Zeiteinheit am größten sind. Spears folgert, dass die Annäherung an das Equilibrium langsamer wird, wenn man für P_0 einen anderen Wert als 0.5 wählt (vgl. Spears 1998, S. 102). Warum dies unabhängig von der Größe der Zugewinne so ist (die ja ebenfalls Einfluss auf die Anteile der Strings haben), erläutert er nicht.

Zugewinne

Damit aus zwei Hyperebenen H_m und H_n die Hyperebene H_k (und damit der String S_i) konstuiert wird, müssen H_m und H_n die Rekombination überleben und sich danach im gleichen Individuum wiederfinden. Dies ist in zwei Fällen möglich: Entweder werden die m Positionen der Hyperebene H_m zwischen den Individuen ausgetauscht und die restlichen Positionen bleiben unverändert. Dann "wandern" die m Allele von H_m zu den n Allelen von H_n :

$$\begin{array}{l} H_m : \mathbf{A} \overbrace{\mathbf{AAA}}^m \mathbf{BB A} \\ H_n : \mathbf{A} \mathbf{BBB} \underbrace{\mathbf{AA}}_n \mathbf{A} \end{array} \rightarrow H_k : \mathbf{ABBBBA} \quad \mathbf{AAAAAA}$$

⁵Spears bezeichnet diesen Fall mit $P_{\text{eq}} = 0.0$ (Spears 1998, S. 31, Gleichung 2.14)

⁶Herleitung siehe Anhang A.2

Oder die n Positionen der Hyperebene H_n werden ausgetauscht und die restlichen Positionen bleiben unverändert. Dann “wandern” die n Allele von H_n zu den m Allelen von H_m .

$$\begin{array}{l} H_m : A \overbrace{AAA}^m BBA \\ H_n : A BBB \underbrace{AAA}_n A \end{array} \rightarrow \begin{array}{l} H_k : AAAAAA \\ \quad \quad \quad ABBBBBA \end{array}$$

Insgesamt geschieht dies mit der Wahrscheinlichkeit

$$P_c(H_k, P_0 | H_m \wedge H_n) = P_0^m (1 - P_0)^n + P_0^n (1 - P_0)^m \quad (4.3)$$

Für welchen Wert von P_0 wird diese Konstruktionswahrscheinlichkeit (und damit der erwartete Zugewinn) maximal? Die Ableitung von P_c ist Null für $P_0 = 0.5$; für $n = m$ befindet sich an dieser Stelle auch das globale Maximum (siehe Anhang A.3). Allerdings können *alle beliebigen* Kombinationen von n und m mit $n + m = k$ auftreten, so dass dies nur ein Spezialfall ist. Für $m = 1$ und $n = 4$ wird die maximale Konstruktionswahrscheinlichkeit z.B. bei $P_0 \approx 0.211$ erreicht (siehe Anhang A.3).⁷

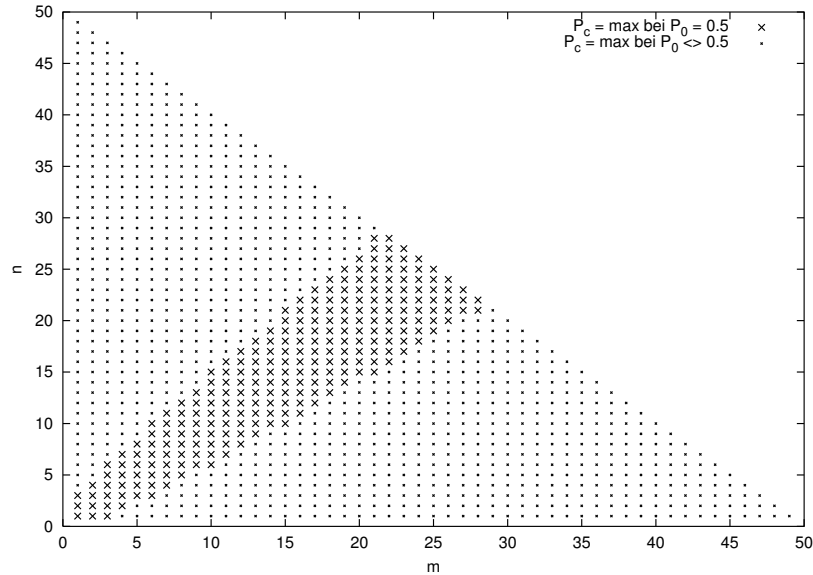


Abbildung 4.1.: Kombinationen von m und n , für die $P_0 = 0.5$ die maximale Konstruktionswahrscheinlichkeit ergibt ($k = m + n \leq 50$)

In Abbildung 4.1 ist aufgetragen, für welche Kombinationen von m und n der Wert $P_0 = 0.5$ die maximale Konstruktionswahrscheinlichkeit liefert.⁸ Es sind alle möglichen Kombinationen für m und n mit $n + m \leq 50$ dargestellt, so dass Aussagen über Hyperebenen bis zur Ordnung 50 möglich sind bzw. über Strings, die sich an maximal 50 Positionen von ihrem Rekombinationspartner unterscheiden. Die Konstruktionswahrscheinlichkeit scheint demnach für $P_0 = 0.5$ auch dann ihr Maximum

⁷Oder für $P_0 = 1 - 0.211 = 0.789$, denn Gleichung 4.3 liefert identische Werte für P_0 und $1 - P_0$.

⁸Die Maxima wurden numerisch anhand von Gleichung 4.3 ermittelt.

zu erreichen, wenn m und n zwar nicht genau gleich, aber ungefähr gleich sind. Dies entspricht dem Bereich rund um die 45-Grad-Achse in Abbildung 4.1.

Nur im Fall $k < 5$ liefert $P_0 = 0.5$ für *alle* möglichen Kombinationen von m und n das globale Maximum. Mit anderen Worten: Für Hyperebenen der Ordnungen 2, 3 und 4 ist die Wahrscheinlichkeit, durch P_0 -Uniform-Rekombination konstruiert zu werden, dann am größten, wenn P_0 gleich 0.5 gewählt wird.⁹ Spears folgert daraus, dass für $k < 5$ das Equilibrium mit $P_0 = 0.5$ am schnellsten erreicht wird. Er unterstützt dies mit einem Beispielversuch, in dem die Veränderungen der Anteile einer Hyperebene der Ordnung 4 für verschiedene Werte von P_0 beobachtet wurden (vgl. Spears 1998, S. 103). Die ursprüngliche Absicht bestand jedoch darin, die Veränderungen der Anteile von *Strings* zu untersuchen. Hyperebenen wurden nur untersucht, da man sich bei der Analyse auf die unterschiedlichen Positionen der beiden Elternstrings beschränken konnte. Daher konnte auch vorausgesetzt werden, dass sich die zu rekombinierenden Hyperebenen an allen Stellen unterscheiden ($P_{eq} = 0.0$).

Wie können die Ergebnisse auf Strings übertragen werden? Unproblematisch sind Strings mit einer Länge kleiner als 5 – wenn sie konstruiert werden, können sich die beiden Eltern nicht an mehr als 4 Stellen unterscheiden, d.h. die zu konstruierende Hyperebene hat eine Ordnung kleiner als 5 und die maximale Konstruktionswahrscheinlichkeit wird für $P_0 = 0.5$ erreicht. Die Zugewinnterme werden dann ebenfalls maximal. Dass dies aber nicht unbedingt die schnellste Annäherung an Robbin's Equilibrium bedeutet, weil dafür die *Differenz* zwischen Zugewinnen und Verlusten entscheidend ist, wurde schon im letzten Abschnitt angemerkt.

4.2.2. Spezialfall: Hyperebenen der Ordnung 2

Als Standard-Rekombinationsoperatoren haben sich P_0 -Uniform-Rekombination und n -point-Rekombination etabliert. Welcher Operator führt einen String schneller ins Equilibrium? Wenn man nicht beliebige Strings, sondern nur Hyperebenen der Ordnung 2 betrachtet, lässt sich diese Frage wie folgt beantworten (vgl. Spears 1998, S. 104ff):

Zunächst soll nur der Fall $C = 2$ (mit $\mathcal{A} = \{A, B\}$) betrachtet werden. Hier sind die vier Hyperebenen $\#A\#A\#$, $\#A\#B\#$, $\#B\#A\#$ und $\#B\#B\#$ von Interesse.¹⁰ Die entsprechenden Differentialgleichungen lauten:

$$\frac{dp_{\#A\#A\#}^{(t)}}{dt} = -p_{\#A\#A\#}^{(t)} p_{\#B\#B\#}^{(t)} P_d(H_2) + p_{\#A\#B\#}^{(t)} p_{\#B\#A\#}^{(t)} P_c(H_2|H_1 \wedge H_1)$$

$$\frac{dp_{\#A\#B\#}^{(t)}}{dt} = -p_{\#A\#B\#}^{(t)} p_{\#B\#A\#}^{(t)} P_d(H_2) + p_{\#A\#A\#}^{(t)} p_{\#B\#B\#}^{(t)} P_c(H_2|H_1 \wedge H_1)$$

$$\frac{dp_{\#B\#A\#}^{(t)}}{dt} = -p_{\#B\#A\#}^{(t)} p_{\#A\#B\#}^{(t)} P_d(H_2) + p_{\#A\#A\#}^{(t)} p_{\#B\#B\#}^{(t)} P_c(H_2|H_1 \wedge H_1)$$

⁹Dies gilt nur unter der getroffenen Annahme, dass die Eltern sich an allen definierenden Positionen der zu konstruierenden Hyperebene unterscheiden.

¹⁰Hierbei handelt es sich um ein Beispiel. Die Länge der Strings (hier: $L = 5$ sowie die definierenden Positionen (hier: 2 und 4) spielen für die folgende Argumentation keine Rolle.

$$\frac{dp_{\#B\#B\#}^{(t)}}{dt} = -p_{\#B\#B\#}^{(t)} p_{\#A\#A\#}^{(t)} P_d(H_2) + p_{\#A\#B\#}^{(t)} p_{\#B\#A\#}^{(t)} P_c(H_2|H_1 \wedge H_1)$$

Die Hyperebene $\#A\#A\#$ kann z.B. nur verloren gehen, wenn sie mit der Hyperebene $\#B\#B\#$ rekombiniert wird, da nur diese sich an $k = 2$ Positionen von ihr unterscheidet. Mit welcher Wahrscheinlichkeit die Hyperebene tatsächlich zerstört wird, hängt von der Disruptionswahrscheinlichkeit des benutzten Rekombinationsoperators in bezug auf Hyperebenen der Ordnung 2 ab. Zugewinne an Hyperebenen $\#A\#A\#$ können nur entstehen, wenn die Hyperebenen $\#A\#B\#$ und $\#B\#A\#$ rekombiniert werden. Mit einer gewissen Wahrscheinlichkeit $P_c(H_2|H_1 \wedge H_1)$ wird dann die Hyperebene $\#A\#A\#$ aus den passenden Hyperebenen der Ordnung 1 ($\#A\#\#\#$ und $\#\#\#A\#$) konstruiert. Entsprechend ergeben sich auch die anderen drei Differentialgleichungen.

Die Disruptions- und Konstruktionswahrscheinlichkeiten sind in den Differentialgleichungen unabhängig von einem bestimmten Rekombinationsverfahren angegeben. In den Fällen, wo zwei Rekombinationsoperatoren sowohl für $P_d(H_2)$ als auch für $P_c(H_2|H_1 \wedge H_1)$ identische Werte besitzen, nähern sich sich Robbings' Equilibrium mit der gleichen Rate.

Beispiel: Vergleich 1-point-Rekombination und P_0 -Uniform-Rekombination

- 1-point-Rekombination: Nach Voraussetzung unterscheiden sich die betrachteten Hyperebenen an allen Positionen (siehe Kapitel 4.2). Die Hyperebene wird deshalb immer dann zerstört, wenn der Schnittpunkt zwischen die beiden definierenden Positionen fällt: $P_d(H_2) = L_1/L$. Gleiches gilt für die Konstruktion; die Hyperebene wird nur dann konstruiert, wenn der Schnittpunkt zwischen die beiden definierenden Positionen fällt (ansonsten bleiben beide Hyperebenen unverändert und es findet keine Konstruktion statt): $P_c(H_2|H_1 \wedge H_1) = L_1/L$
- P_0 -Uniform-Rekombination: Da sich die betrachteten Hyperebenen an allen Positionen unterscheiden, wird die Hyperebene in fast allen Fällen zerstört. Überleben kann sie nur, wenn alle k Gene ausgetauscht werden oder gar keines von ihnen: $P_d = 1 - (1 - P_0)^2 - P_0^2 = 2P_0(1 - P_0)$. Damit die gewünschte Hyperebene der Ordnung 2 konstruiert wird, muss genau eines der beiden Allele zwischen den beiden Hyperebenen ausgetauscht werden, d.h. entweder das erste wird ausgetauscht und das zweite nicht oder das zweite wird ausgetauscht und das erste nicht:¹¹ $P_c(H_2|H_1 \wedge H_1) = (1 - P_0)P_0 + P_0(1 - P_0) = 2P_0(1 - P_0)$

Für den Fall $L_1/L = 2P_0(1 - P_0) \Leftrightarrow L_1 = 2LP_0(1 - P_0)$ sind die Disruptions- und Konstruktionswahrscheinlichkeiten von 1-point-Rekombination und P_0 -Uniform-Rekombination gleich groß. In diesem Fall verhalten sich die beiden Operatoren in Bezug auf Hyperebenen der Ordnung 2 identisch.

Bisher wurde die Argumentation nur für $C = 2$ geführt. Sie bleibt jedoch auch für größere Alphabete ($C > 2$) gültig. Dann erhält man zwar mehr Differentialgleichungen und für die Verlust- und Zugewinnterme jeweils mehrere Summanden. Aber es gehen wieder die gleichen Disruptions- bzw. Konstruktionsterme ein wie für $C = 2$ (vgl. Spears 1998, S. 106).

¹¹siehe Gleichung 4.3 mit $n = m = 1$ (vgl. Spears 1998, Gleichung 3.6, S. 44)

5. Analyse der Kombination von Mutation und Rekombination

5.1. Art des Equilibriums

Werden Rekombination und Mutation fortlaufend hintereinander auf eine Population angewandt, stellt sich die Frage, welchem Gleichgewichtszustand sich die Population annähert: dem Uniform-Equilibrium (wie bei ausschließlicher Mutation) oder Robbins' Equilibrium (wie bei ausschließlicher Rekombination)?

Obwohl sich die beiden Gleichgewichte voneinander unterscheiden, strebt die Population in gewissem Sinne beide Equilibrii an: Nach wie vor verursacht die Rekombination eine Annäherung an Robbins' Equilibrium. Dieses wird bestimmt von den Anteilen der Allele. Während bei alleiniger Rekombination diese Anteile konstant bleiben, werden sie hier durch die Mutation immer mehr einander angeglichen. Auf diese Weise verändert die Mutation das von der Rekombination angestrebte Robbins' Equilibrium mit der Zeit immer mehr in Richtung des Uniform Equilibrium (Spears 1998, S. 113):

$$\lim_{t \rightarrow \infty} p_S^{(t)} = \prod_{i=1}^L \frac{1}{C} \quad (5.1)$$

5.2. Annäherungsgeschwindigkeit an das Equilibrium

Es stellt sich die Frage, ob die Annäherungsgeschwindigkeit an das Uniform Equilibrium durch die in jedem Zeitabschnitt zusätzlich erfolgende Annäherung an das jeweilige Robbin's Equilibrium verändert wird. Spears stellt nur fest, dass eine höhere Mutationrate nach wie vor eine höhere Annäherungsgeschwindigkeit zur Folge habe (vgl. Spears 1998, S. 113).

Die Herleitung von Gleichung 3.10 im Kapitel Mutation basierte auf einer Differentialgleichung, in die neben den Konstanten μ und C nur $p_\alpha^{(t)}$ (der Anteil des untersuchten Allels an der Population zum Zeitpunkt t) einging. Der hier nun vorgeschaltete Rekombinationsoperator kann jedoch diese Anteile nicht verändern. Daher bleibt die Differentialgleichung unverändert gültig. Somit lässt sich vermuten, dass sich für die Kombination von Mutation und Rekombination die gleiche Annäherungsgeschwindigkeit an das Uniform Equilibrium ergibt wie bei ausschließlicher Mutation:

$$p_\alpha^{(t)} = \frac{1}{C} + \left(p_\alpha^{(0)} - \frac{1}{C} \right) e^{\frac{-C\mu t}{C-1}} \quad (5.2)$$

6. Zusammenfassung und Ausblick

In dieser Arbeit wurde untersucht, wie sich Populationen verändern, wenn sie den genetischen Operatoren Mutation und Rekombination ausgesetzt werden.

Eine Population die ausschließlich *mutiert* wird, nähert sich exponentiell einem Uniform Equilibrium an, in dem alle Strings gleich wahrscheinlich sind. Im wesentlichen hängt es von den Anfangsbedingungen und der Mutationsrate ab, wie schnell das Uniform Equilibrium erreicht wird; einen kleinen Einfluss besitzt auch das gewählte Alphabet.

Werden die Individuen einer Population immer wieder *rekombiniert*, nähert sich die Population Robbins' Equilibrium an. Diese Verteilung der Anteile der Strings wird nach Geiringers Theorem nur von der Allelverhältnissen an den einzelnen Positionen bestimmt; diese bleiben über die Zeit konstant und können daher bereits an der Anfangspopulation abgelesen werden. Die Geschwindigkeit der Annäherung an Robbins' Equilibrium wird im wesentlichen von den statischen Disruptions- und Konstruktionswahrscheinlichkeiten des gewählten Rekombinationsoperators in bezug auf Hyperebenen bestimmt. So ergibt sich ein enger Zusammenhang dieser dynamischen Analyse mit traditionellen *statischen* Analysen. Sind die Disruptions- und Konstruktionswahrscheinlichkeiten maximal, wird das Equilibrium am schnellsten erreicht. Zwei Spezialfälle wurden näher untersucht: Bei P_0 -Uniform-Rekombination liefert $P_0 = 0.5$ anscheinend in vielen Fällen die höchste Annäherungsrate. In Bezug auf Hyperebenen der Ordnung 2 bedeuten gleiche Disruptions- und Konstruktionswahrscheinlichkeiten die gleiche Annäherungsrate an das Equilibrium.

Wenn sowohl Rekombination als auch Mutation zugelassen werden, strebt die Population langfristig das Uniform Equilibrium an. Der Rekombinationsschritt führt zwar wie in der isolierten Betrachtung zu einer Annäherung an ein Robbins' Equilibrium, dieses wird jedoch durch den Mutationsoperator mit der Zeit selbst immer mehr dem Uniform Equilibrium angenähert. Die Geschwindigkeit der Annäherung an das Uniform Equilibrium ist vermutlich die gleiche wie bei ausschließlicher Mutation.

Diese Ergebnisse helfen bei der Charakterisierung der beiden genetischen Operatoren Mutation und Rekombination und erlauben Vergleiche zwischen ihnen. Für eine dynamische Analyse eines vollständigen Genetischen Algorithmus fehlt jedoch der Selektionsoperator. In einem nächsten Schritt gilt es deshalb, die Selektion in diese Analysen miteinzubeziehen.

Die Wiederentdeckung der Ergebnisse Hilda von Geiringers aus dem Jahr 1944 bei der Analyse der Rekombination zeigt, das es Sinn macht, bei der Erforschung Genetischer Algorithmen die Ergebnisse der Populationsgenetik zu berücksichtigen. Mühlenbein sagt dazu: "Die mathematische Populationsgenetik hat eine wichtige Rolle beim Verständnis der Evolution gespielt. Sie wird eine ebenso wichtige Rolle bei der Entwicklung einer Theorie Genetischer Algorithmen spielen."(Mühlenbein 1995)

A. Nebenrechnungen

A.1. Berechnung der Annäherung an das Uniform Equilibrium

Zu lösen ist die Gleichung 3.9:

$$\frac{dp}{dt} = \frac{\mu}{C-1} (1 - Cp) \quad \text{mit} \quad 0 \leq \mu \leq 1 \quad \text{und} \quad C \in \mathbb{N} \wedge C > 1 \wedge Cp \neq 1.$$

Es handelt sich hierbei um eine gewöhnliche lineare Differentialgleichung erster Ordnung für die Funktion $p = p(t)$. Zur Lösung wendet man die *Trennung der Variablen* an (vgl. Forster 1984, S. 111).

$$\frac{dp}{1 - Cp} = \frac{\mu}{C-1} dt.$$

Integration: auf der linken Seite von $p(0) \equiv p_0$ bis $p(t)$ und auf der rechten Seite von 0 bis t :

$$\int_{p_0}^{p(t)} \frac{d\tilde{p}}{1 - C\tilde{p}} = \int_0^t \frac{\mu}{C-1} d\tilde{t}$$

Da μ und C konstant sind:

$$\begin{aligned} \int_{p_0}^{p(t)} \frac{d\tilde{p}}{1 - C\tilde{p}} &= \frac{\mu}{C-1} \int_0^t d\tilde{t} \\ \int_{p_0}^{p(t)} \frac{d\tilde{p}}{1 - C\tilde{p}} &= \frac{\mu}{C-1} t \end{aligned}$$

Das Integral über p auf der linken Seite hat die Form $\int \frac{dx}{1-ax}$. Die Stammfunktion dieses Integrals lautet $-\frac{1}{a} \ln(1 - ax) + \text{Konstante}$ (vgl. Luderer u. a. 1998). Damit ergibt sich:

$$\begin{aligned} \left[-\frac{1}{C} \ln(1 - C\tilde{p}) \right]_{p_0}^{p(t)} &= \frac{\mu}{C-1} t \\ -\frac{1}{C} \ln(1 - Cp(t)) + \frac{1}{C} \ln(1 - Cp_0) &= \frac{\mu}{C-1} t \end{aligned}$$

Zusammenfassung der Logarithmen:

$$-\frac{1}{C} \ln \frac{1 - Cp(t)}{1 - Cp_0} = \frac{\mu}{C-1} t$$

Multiplikation beider Seiten mit $-C$:

$$\ln \frac{1 - Cp(t)}{1 - Cp_0} = -\frac{C\mu}{C-1} t$$

Exponentieren:

$$\frac{1 - Cp(t)}{1 - Cp_0} = e^{-\frac{C\mu}{C-1}t}$$

Multiplikation mit $1 - Cp_0$:

$$1 - Cp(t) = (1 - Cp_0) e^{-\frac{C\mu}{C-1}t}$$

Subtraktion von 1:

$$-Cp(t) = -1 + (1 - Cp_0) e^{-\frac{C\mu}{C-1}t}$$

Division durch $-C$:

$$p(t) = \frac{1}{C} + \left(p_0 - \frac{1}{C}\right) e^{-\frac{C\mu}{C-1}t}$$

A.2. Bestimmung des globalen Maximums für die Disruptionswahrscheinlichkeit

Gesucht ist das globale Maximum der folgenden Funktion (siehe Gleichung 4.2):

$$P_d(P_0) = 1 - P_0^k - (1 - P_0)^k \quad \text{für } P_0 \in [0; 1], k \in \mathbb{N}, k \geq 2$$

Nullsetzen der ersten Ableitung:

$$\frac{dP_d(P_0)}{dP_0} = -kP_0^{k-1} + k(1 - P_0)^{k-1} \stackrel{!}{=} 0$$

$$(1 - P_0)^{k-1} = P_0^{k-1}$$

$$1 - P_0 = P_0$$

$$P_0 = 0.5$$

Zweite Ableitung:

$$\frac{d^2P_d(P_0)}{d^2P_0} = -k(k-1)P_0^{k-2} - k(k-1)(1 - P_0)^{k-2} < 0$$

Die zweite Ableitung ist für alle zulässigen Werte von k und P_0 negativ. Also befindet sich an der Stelle $P_0 = 0.5$ ein lokales Maximum. Überprüfung der Randstellen:

$$P_d(0) = 1 - 0^k - (1 - 0)^k = 0$$

$$P_d(1) = 1 - 1^k - (1 - 1)^k = 0$$

$$P_d(0.5) = 1 - 0.5^k - (1 - 0.5)^k = 1 - 2 \cdot 0.5^k > 0$$

Damit liegt an der Stelle $P_0 = 0.5$ ein globales Maximum der Funktion P_d vor.

A.3. Bestimmung des globalen Maximums für die Konstruktionswahrscheinlichkeit

Gesucht ist das globale Maximum der Funktion P_c (siehe Gleichung 4.3):

$$P_c(P_0) = P_0^m(1 - P_0)^n + P_0^n(1 - P_0)^m \quad \text{für } n, m \in \mathbb{N}, P_0 \in [0; 1]$$

Ableiten nach P_0 durch Anwendung der Produktregel:

$$\frac{dP_c}{dP_0} = mP_0^{m-1}(1 - P_0)^n - P_0^n(1 - P_0)^{n-1} + nP_0^{n-1}(1 - P_0)^m - P_0^m(1 - P_0)^{m-1} \stackrel{!}{=} 0$$

Sortieren der Terme ergibt die Bedingung für die Nullstelle der ersten Ableitung:

$$mP_0^{m-1}(1 - P_0)^n + nP_0^{n-1}(1 - P_0)^m = m(1 - P_0)^{m-1}P_0^n + n(1 - P_0)^{n-1}P_0^m$$

An der Symmetrie der Gleichung erkennt man, dass sie für $P_0 = 1 - P_0 \Leftrightarrow P_0 = 0.5$ erfüllt ist, d.h. an $P_0 = 0.5$ hat P_c ein lokales Extremum.

Für $n = m$ ist P_0 ein lokales Maximum, wie sich im Folgenden zeigen lässt. Die erste Ableitung wird für $n = m$ zu

$$mP_0^{m-1}(1 - P_0)^m - mP_0^m(1 - P_0)^{m-1} + mP_0^{m-1}(1 - P_0)^m - mP_0^m(1 - P_0)^{m-1}$$

was sich vereinfachen lässt zu

$$2mP_0^{m-1}(1 - P_0)^m - 2mP_0^m(1 - P_0)^{m-1}$$

Falls an der Stelle $P_0 = 0.5$ ein Maximum vorliegt, ist die erste Ableitung für $P_0 < 0.5$ größer als Null.

$$2mP_0^{m-1}(1 - P_0)^m - 2mP_0^m(1 - P_0)^{m-1} \stackrel{!}{>} 0$$

$$2mP_0^{m-1}(1 - P_0)^m > 2mP_0^m(1 - P_0)^{m-1}$$

$$\frac{(1 - P_0)^m}{(1 - P_0)^{m-1}} > \frac{P_0^m}{P_0^{m-1}}$$

$$1 - P_0 > P_0$$

$$P_0 < 0.5$$

Analog zeigt man, dass die erste Ableitung für $P_0 > 0.5$ kleiner als Null ist. Damit liegt für $n = m$ an der Stelle $P_0 = 0.5$ ein lokales Maximum vor. Da $P_c(P_0)$ für beide Randwerte ($P_0 = 0$ und $P_0 = 1$) Null wird, für $P_0 = 0.5$ aber echt positiv, handelt es sich nicht nur um ein lokales, sondern auch um ein globales Maximum.

Für $n \neq m$ ist das lokale Extremum bei $P_0 = 0.5$ nicht immer ein globales Maximum, wie sich durch ein Gegenbeispiel für $m = 1$ und $n = 4$ zeigen lässt:

$$P_c(0.5) = 0.5(1 - 0.5)^4 + 0.5^4(1 - 0.5) = 0.0625$$

Das globale Maximum wird in diesem Fall ungefähr an den Stellen $P_0 = 0.211$ und $P_0 = 1 - 0.211 = 0.789$ angenommen:

$$P_c(0.211) = P_c(1 - 0.211) = 0.211(1 - 0.211)^4 + 0.211^4(1 - 0.211) \approx 0.0833$$

Tatsächlich liegt an der Stelle $P_0 = 0.5$ ein lokales Minimum vor, wie man an Abbildung A.1 erkennt.

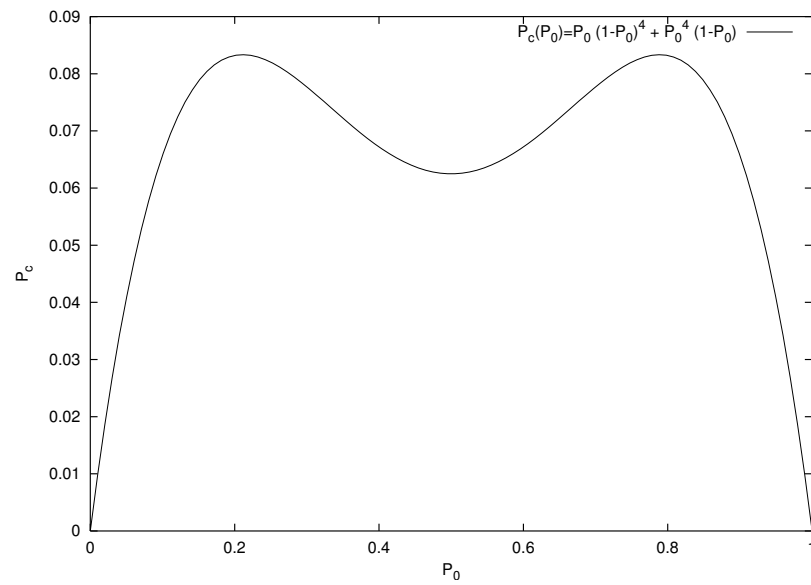


Abbildung A.1.: Konstruktionswahrscheinlichkeit P_c in Abhängigkeit von P_0 , für $m = 1$ und $n = 4$

Literatur

- [Forster 1984] FORSTER, Otto: *Analysis 2*. 5. durchges. Aufl. Vieweg, 1984
- [Juels u. a. 1993] JUELS, Ari ; BALUJA, Shumeet ; SINCLAIR, Alistair: *The Equilibrium Genetic Algorithm and the Role of Crossover*. 1993. – URL: http://www.dai.ed.ac.uk/groups/evalg/Local_Copies_of_Papers/Juels.Baluja.Sinclair.The_Equilibrium_Genetic_Algorithm_and_the_Role_of_Crossover.ps.gz (2001-11-22)
- [Luderer u. a. 1998] LUDERER, Bernd ; NOLLAU, Volker ; VETTERS, Klaus: *Mathematische Formeln für Wirtschaftswissenschaftler*. Stuttgart : B. G. Teubner, 1998
- [Mühlenbein 1995] MÜHLENBEIN, Heinz: Genetische Algorithmen und Evolutionstheorien: Auf der Suche nach verschollenen Schätzen. In: *GMD Spiegel 2'95* (1995). – URL: <http://set.gmd.de/AS/gmdsp/muehlen.html> (2001-11-22)
- [Mühlenbein und Mahnig 2000] MÜHLENBEIN, Heinz ; MAHNIG, Thilo: Evolutionary Algorithms: From Recombination to Search Distributions. In: KALLEL, L. (Hrsg.) ; NAUDTS, B. (Hrsg.) ; ROGERS, A. (Hrsg.): *Theoretical Aspects of Evolutionary Computing*, Springer Verlag, 2000 (Natural Computing). – URL: <http://ais.gmd.de/~mahnig/publications/MueMa00c.pdf> (2001-11-22), S. 137–176
- [Nissen 1994] NISSEN, Volker: *Evolutionäre Algorithmen: Darstellung, Beispiele, betriebswirtschaftliche Anwendungsmöglichkeiten*. Wiesbaden : Deutscher Universitäts-Verlag, 1994
- [Pohlheim 1999] POHLHEIM, Hartmut: *Evolutionäre Algorithmen: Verfahren, Operatoren und Hinweise für die Praxis*. Berlin : Springer, 1999
- [Spears 1998] SPEARS, William M.: *The Role of Mutation and Recombination in Evolutionary Algorithms*. Fairfax, Virginia, George Mason University, Diss., 1998. – URL: <http://www.cs.uwo.edu/~wspears/papers/thesis.double.ps> (2001-11-22)
- [Whitley 1994] WHITLEY, Darrell: A Genetic Algorithm Tutorial. In: *Statistics and Computing* 4 (1994), S. 65–85. – URL: ftp://ftp.cs.colostate.edu/pub/public_html/TechReports/1993/tr-103.ps.Z (2001-11-22)