

---

# Genetische Algorithmen

---

Projektgruppe 431 — Metaheuristiken

Bianca Selzam

# Inhaltsverzeichnis

1	Einleitung .....	1
2	Grundlagen aus der Biologie .....	2
3	Grundbegriffe aus der Genetik .....	2
3.1	Individuum / Chromosom .....	2
3.2	Gen .....	2
3.3	Allel .....	3
3.4	Länge eines Chromosoms .....	3
3.5	Genotyp .....	3
3.6	Phänotyp .....	3
3.7	Population, Generation .....	3
4	Mathematische Grundlagen .....	4
4.1	Minimum, Maximum .....	4
4.2	Zufallsvariable .....	4
4.3	Wahrscheinlichkeitsverteilung .....	4
4.4	Verteilungsfunktion .....	4
4.5	Erwartungswert .....	5
4.6	Varianz, Standardabweichung .....	5
4.7	Normalverteilung .....	6
5	Aufbau eines genetischen Algorithmus .....	7
5.1	Kodierung .....	7
5.2	Fitness- und Bewertungsfunktion .....	8
5.3	Selektion .....	8
5.4	Rekombination/Kreuzung .....	10
5.5	Mutation .....	11
5.6	Ersetzungsschema .....	12
5.7	Abbruchkriterium .....	12
6	Anwendung eines genetischen Algorithmus .....	13
7	Zusammenfassung .....	14
8	Glossar .....	15
	Literatur .....	17

# 1 Einleitung

Evolutionäre Algorithmen (EA) sind Optimierungsverfahren, die sich am Vorbild der biologischen Evolution orientieren. Sie lassen sich prinzipiell in vier Teilgebiete unterteilen:

- Genetische Algorithmen (GA)
- Genetische Programmierung (GP)
- Evolutionsstrategien (ES)
- Evolutionäre Programmierung (EP)

Genetische Algorithmen wurden erstmals Anfang der 60er Jahre von John Holland und seinen Mitarbeitern der University of Michigan vorgestellt [2]. Etwa zur gleichen Zeit begründeten Ingo Rechenberg und Hans-Paul Schwefel an der TU Berlin die Evolutionsstrategien [4, 6]. Die Unterteilung in die einzelnen Teilgebiete ist somit hauptsächlich geschichtlich bedingt, da sie sehr hohe Ähnlichkeiten aufweisen.

Dennoch existieren zwischen den Teilgebieten einige inhaltliche bzw. den Schwerpunkt der Gebiete betreffende Unterschiede. Beispielsweise werden in den genetischen Algorithmen die Individuen als Bitstrings kodiert, während bei den Evolutionsstrategien und der Evolutionären Programmierung meistens reelle Vektoren verwendet werden.

Ein weiterer Unterschied zwischen den Teilgebieten besteht in der Selektion der Individuen. Genetische Algorithmen und Evolutionäre Programmierung nehmen eine stochastische Selektion vor, d. h. auch die Individuen mit einem schlechten Fitness-Wert haben eine, wenn auch geringe, Chance auf Weitergabe ihrer Gene an die nächste Generation. Bei den Evolutionsstrategien dagegen wird eine deterministische Selektion vorgenommen, d. h. nur die Individuen mit der besten Fitness geben ihre Gene weiter.

In den folgenden Kapiteln soll näher auf die biologischen Hintergründe, mathematischen Grundlagen sowie Eigenschaften und Anwendungsgebiete der genetischen Algorithmen eingegangen werden.

## 2 Grundlagen aus der Biologie

Jedes Lebewesen trägt in seinen Zellen die charakteristische Erbinformation, die im Allgemeinen im Laufe seines Lebens unveränderlich ist. Auf eine ganze Gattung gesehen, verändern sich jedoch diese Informationen im Lauf der Zeit.

Bei der Fortpflanzung von Organismen entstehen neue Baupläne. Paaren sich beispielsweise zwei Tiere, so werden ihre individuellen Baupläne miteinander kombiniert und zu neuen zusammengesetzt. Deshalb sind die gezeugten Kinder ihren beiden Elternteilen ähnlich, jedoch nie mit ihnen identisch.

Weiterhin können während der Fortpflanzung spontane Änderungen der Baupläne auftreten, die ebenfalls zu neuen Erbinformationen führen. Solche Änderungen werden z. B. durch äußere Umwelteinflüsse verursacht.

Das Evolutionsprinzip „Survival of the fittest“ nach Charles Darwin legt die Theorie zugrunde, dass diejenigen Lebewesen einer Gattung überleben, die sich am besten an die vorherrschenden äußeren Umweltbedingungen angepasst haben. Diese Organismen haben somit die größte Zahl an Nachkommen und beeinflussen die weitere Entwicklung ihrer Gattung am entscheidendsten. Man bezeichnet dieses Phänomen oft als „natürliche Selektion“, da sozusagen die Natur eine Auswahl an Lebewesen trifft, deren Nachkommen immer besser an ihre Umgebungen angepasst sind.

## 3 Grundbegriffe aus der Genetik

### 3.1 Individuum / Chromosom

Ein **Individuum** im biologischen Sinne ist ein lebender Organismus, dessen Erbinformationen in einer Menge von Chromosomen gespeichert ist. Im Zusammenhang mit genetischen Algorithmen werden die Begriffe Individuum und **Chromosom** jedoch meistens gleichgesetzt.

Ein Individuum wird als Binärstring der festen Länge  $n$  kodiert, d. h. man kann es als Element aus  $\{0, 1\}^n$  auffassen.

### 3.2 Gen

Eine bestimmte Stelle bzw. Sequenz eines Chromosoms wird als **Gen** bezeichnet. Üblicherweise geht aus dem Kontext hervor, ob man unter einem Gen eine einzelne Stelle oder einen ganzen Abschnitt versteht.

### 3.3 Allel

Die konkrete Ausprägung eines Gens wird als **Allel** bezeichnet. Wird das Gen als Variable aufgefasst, so ist das Allel der Wert der Variablen. Wenn die Gene einzelne Stellen des Binärstrings bezeichnen, so können die Allele nur die Werte 0 und 1 annehmen.

### 3.4 Länge eines Chromosoms

Unter der **Länge eines Chromosoms** versteht man die Länge des Binärvektors, d. h. die Anzahl der Gene eines Individuums.

### 3.5 Genotyp

Der **Genotyp** ist der kodierte Vektor der Entscheidungsvariablen. Von ihm hängt im Allgemeinen ab, welche Kodierungsmethode gewählt wird.

### 3.6 Phänotyp

Der **Phänotyp** dagegen ist der dekodierte Vektor der Entscheidungsvariablen. Seine Ausprägung hängt vom Genotyp und der gewählten Dekodierungsmethode ab.

### 3.7 Population, Generation

Eine Menge von strukturell gleichartigen Individuen einer bestimmten Gattung wird als **Population** bezeichnet. Wenn neue Lebewesen dieser Gattung geboren werden oder andere sterben, verändert sich zwangsläufig auch die Größe der Population. Betrachtet man die Populationen einer Gattung über mehrere Zeitpunkte hinweg, spricht man von **Generationen** der Lebewesen.

## 4 Mathematische Grundlagen

### 4.1 Minimum, Maximum

Eine Zahl  $a \in \mathcal{M}$  heißt **Minimum** von  $\mathcal{M}$ , falls für alle  $x \in \mathcal{M}$  gilt:  $a \leq x$ . Man bezeichnet das Minimum von  $\mathcal{M}$  mit  $\min(\mathcal{M})$ .

Eine Zahl  $a \in \mathcal{M}$  heißt **Maximum** von  $\mathcal{M}$ , falls für alle  $x \in \mathcal{M}$  gilt:  $a \geq x$ . Man bezeichnet das Maximum von  $\mathcal{M}$  mit  $\max(\mathcal{M})$ .

Minima und Maxima sind eindeutig bestimmt. Falls zwei Minima bzw. Maxima  $a_1$  und  $a_2$  existieren, so gilt  $a_1 = a_2$ .

Wenn eine Menge  $\mathcal{M}$  ein Maximum bzw. Minimum besitzt, so ist  $\min(\mathcal{M})$  bzw.  $\max(\mathcal{M})$  das kleinste bzw. größte Element von  $\mathcal{M}$ .

### 4.2 Zufallsvariable

Eine Variable, deren Werte durch Zufallsereignisse bestimmt oder verändert werden, wird als **Zufallsvariable** bezeichnet. Kann die Variable nur bestimmte Werte annehmen, spricht man von einer **diskreten**, andernfalls von einer **stetigen** Zufallsvariable.

### 4.3 Wahrscheinlichkeitsverteilung

Die Funktion, die für jede Ausprägung der Zufallsvariable die Wahrscheinlichkeit ihres Auftreten angibt, wird Wahrscheinlichkeitsfunktion bzw. **Wahrscheinlichkeitsverteilung**  $f$  der Zufallsvariable  $X$  genannt:

$$f(x_i) = W(X = x_i) \quad (1)$$

Eine Wahrscheinlichkeitsfunktion besitzt die folgenden Eigenschaften:

$$f(x_i) \geq 0 \quad (2)$$

$$\sum_i f(x_i) = 1 \quad (3)$$

### 4.4 Verteilungsfunktion

Mit Hilfe der Wahrscheinlichkeitsfunktion kann die sogenannte **Verteilungsfunktion** definiert werden. Sie gibt die Wahrscheinlichkeit dafür an, dass die Zufallsvariable  $X$  höchstens den Wert  $x$  annimmt:

$$F(x) = W(X \leq x) = \sum_{x_i \leq x} f(x_i) \quad (4)$$

Falls  $X$  eine diskrete Zufallsvariable ist, so ist der Graph der Verteilungsfunktion eine Treppenfunktion.

Andernfalls ist der Graph eine stetige Funktion mit folgenden Eigenschaften:

$$0 \leq F(x) \leq 1 \quad (5)$$

$$x_i < x_j \quad (6)$$

$$f(x_i) \geq 0 \quad (7)$$

$$\sum_i f(x_i) = 1 \quad (8)$$

## 4.5 Erwartungswert

Der **Erwartungswert**  $E(X)$  einer diskreten Zufallsvariable  $X$  ist definiert als

$$E(X) = \sum_i x_i * W(X = x_i), \quad (9)$$

d. h. die Werte der Zufallsvariable werden mit ihren Auftretswahrscheinlichkeiten multipliziert und aufsummiert.

Für stetige Zufallsvariablen  $X$  gilt folgende Gleichung:

$$E(X) = \int_{-\infty}^{+\infty} x * f(x) dx, \quad (10)$$

wobei  $f(x)$  die Ableitung der Verteilungsfunktion  $F(x)$  ist. Sie wird auch als **Dichtefunktion** bezeichnet.

## 4.6 Varianz, Standardabweichung

Die **Varianz**  $Var(X)$  einer diskreten Zufallsvariable  $X$  ist definiert als

$$Var(X) = \sum_i (x_i - E(X))^2 * f(x_i) \quad (11)$$

Für stetige Zufallsvariablen  $X$  gilt folgende Gleichung:

$$Var(X) = \int_{-\infty}^{+\infty} (x - E(X))^2 * f(x) dx. \quad (12)$$

Die Varianz ist ein Maß für die Streuung der Ausprägungen der Zufallsvariable  $X$ .

Ein weiteres Maß für die Streuung der Werte von  $X$  ist die **Standardabweichung**. Sie ist definiert als die positive Wurzel der Varianz:

$$s(X) = \sqrt{\text{Var}(X)} \quad (13)$$

## 4.7 Normalverteilung

Die Normalverteilung ist die wichtigste statistische Verteilung, da sehr viele Eigenschaftsausprägungen in der Natur normalverteilt auftreten. Außerdem können viele weitere Verteilungen mit Hilfe der Normalverteilung einfach angenähert werden.

Die Dichtefunktion der Normalverteilung hat die Gestalt einer Glocke und wird daher auch als Gauß'sche Glockenkurve bezeichnet. Die konkrete Gestalt der Kurve ist von drei Parametern abhängig:

- die Ausprägungen der stetigen Variable  $X$  und ihre Auftretswahrscheinlichkeiten
- der Erwartungswert  $m$
- die Standardabweichung  $s$ .

Die Funktionsgleichung der Gauß'schen Glockenkurve lautet:

$$f(x, m, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} * e^{-\frac{(x-m)^2}{2\sigma^2}} \quad (14)$$

Hierbei bestimmt  $m$  die Lage der Glockenkurve und  $s$  die Streuung. Das Maximum der Dichtefunktion liegt bei  $x = m$ , die Wendepunkte der Glockenkurve liegen bei  $m - s$  und  $m + s$ .

Das Integral der Gauß'schen Glockenkurve stellt die Verteilungsfunktion der Normalverteilung dar. Sie hat eine S-förmige Gestalt, und ihre Formel lautet:

$$F(x, m, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} * \int_{-\infty}^x e^{-\frac{(t-m)^2}{2\sigma^2}} dt \quad (15)$$



## 5 Aufbau eines genetischen Algorithmus

Genetische Algorithmen können üblicherweise in folgende Subroutinen aufgeteilt werden:

1. Das zu optimierende Problem wird **kodiert**, d. h. es wird auf ein binär kodiertes Chromosom abgebildet.
2. Eine Population von Individuen wird erzeugt und zufällig initialisiert. Man spricht hier von der **Ausgangspopulation** bzw. Generation 0.
3. Jedes Individuum wird mit einer **Fitnessfunktion** bewertet, welche jedem einzelnen Chromosom eine reellwertige Zahl zuordnet.
4. Jeweils zwei Elternteile werden mittels einer gewählten **Selektionsvariante** selektiert.
5. Aus den genetischen Informationen der Eltern werden mittels einer gewählten **Kreuzungsvariante** die Nachkommen erzeugt.
6. Die Allele der Nachkommen können **mutieren**, d. h. ihre Werte werden invertiert.
7. Die Population wird um die neu erzeugten Nachkommen ergänzt. Wird die Populationsgröße überschritten, wird nach einem gewählten **Ersetzungsschema** eine Menge von Individuen bestimmt, die durch die neuen ersetzt werden.
8. Ab Schritt 4 werden die Subroutinen so lange wiederholt, bis ein **Abbruchkriterium** erfüllt ist.

### 5.1 Kodierung

In genetischen Algorithmen werden die Chromosomen bevorzugt mit dem Gray-Code anstelle des Standard-Binär-codes kodiert. Der Gray-Code hat eine Hamming-Distanz von 1, d. h. zwei aufeinanderfolgende Codeworte unterscheiden sich jeweils in nur einer Binärstelle. Der Standard-Binär-codier hingegen hat unterschiedliche Hamming-Distanzen, wie folgendes Beispiel verdeutlichen soll:

dezimal	Standard-Binär-codier	Gray-Code
0	000	000
1	001 (d=1)	001 (d=1)
2	010 (d=2)	011 (d=1)
3	011 (d=1)	010 (d=1)
4	100 (d=3)	110 (d=1)

d = Hamming-Distanz (k, k-1)

Im Standard-Binär-codier repräsentiert jede Binärstelle eine Zweierpotenz. Durch eine zufällige Mutation an einer höherwertigen Stelle können somit große Sprünge im Phänotyp auftreten. Beispielsweise unterscheiden sich die beiden Genotypen 1000001 und 0000001 nur durch eine Binärstelle, die entsprechenden Phänotypen 65 und 1 liegen jedoch recht weit auseinander. Andererseits beträgt

die Hamming-Distanz zwischen den binären Darstellungen 01111 und 10000 fünf, obwohl sich die zugehörigen Phänotypen 15 und 16 nur um eins unterscheiden.

Durch Verwendung des Gray-Codes wirkt sich eine Mutation in den höherwertigen Bits nicht so gravierend auf die Phänotypen aus. Jedoch gibt es auch Fälle, in denen der Standard-Binärcode eine geringere Veränderung im Phänotyp als der Gray-Code bewirkt. Hierzu ein Beispiel:

Dezimal:	3
Standard-Binärcode:	011
Gray-Code:	010

Nach der Mutation des 2. Bits ergeben sich folgende Werte:

Standard-Binärcode:	001	Gray-Code:	000
Dezimal:	1	Dezimal:	0
Differenz:	2	Differenz:	3

Die mutationsbedingten Veränderungen eines Allels bei Verwendung des Gray-Codes sind also nicht immer geringfügiger als beim Standard-Binärcode.

## 5.2 Fitness- und Bewertungsfunktion

Die Bewertungsfunktion misst die Güte eines Individuums bezüglich der zu optimierenden Aufgabe, während die Fitnessfunktion seine Chancen auf Fortpflanzung bewertet. Man kann beide Funktionen gleichsetzen, wenn die am Optimum gemessenen besten Individuen auch die besten Chancen auf Fortpflanzung haben sollen.

Eine beliebte Fitnessfunktion stellt die sogenannte proportionale oder lineare Fitness dar, welche die Fitness eines Individuums in direkter Proportionalität zur Bewertung stellt:

$$Prop_{fit}(I) := \frac{a * I}{B} \quad (16)$$

I = Bewertung des Individuums, B = Summe der Bewertungen aller Individuen und a = beliebiger Faktor

## 5.3 Selektion

Die Selektion kann unterteilt werden in einen Selektionsschritt und einen Auswahlsschritt. Der Selektionsalgorithmus weist dabei jedem Chromosom einen Wahrscheinlichkeitswert für dessen Replikation zu. Dieser Wert ist zunächst ein Erwartungswert

$$E(I) = \mu * p_s(I), \quad (17)$$

wobei  $\mu$  die Populationsgröße und  $p_s(I)$  die Selektionswahrscheinlichkeit des einzelnen Individuums ist. Somit gibt  $E(I)$  die zu erwartende Anzahl von Replikaten im sogenannten „mating pool“ an.

### 5.3.1 Selektionsalgorithmen

#### Fitnessproportionale Selektion

Bei dieser Selektionsmethode ist die Selektionswahrscheinlichkeit  $p_s(I)$  direkt proportional zur Fitness des jeweiligen Individuums:

$$p_s = \frac{\Phi(I_j)}{\sum_{j=1}^n \Phi(I_j)} \quad (18)$$

$\Phi$  ist dabei die Bewertungsfunktion,  $I$  ein Individuum und  $j$  der Index des Individuums.

Jedoch ist der Selektionsdruck bei dieser Vorgehensweise verhältnismäßig gering. Je höher der Selektionsdruck, desto schneller konvergiert der Algorithmus und desto schneller wird ein lokales Optimum gefunden.

Ein Maß für den Selektionsdruck ist die sogenannte „takeover time“. Sie gibt die Anzahl der Generationen an, die benötigt wird, um mit einem gegebenen Selektionsalgorithmus bei alleiniger Anwendung der Selektion eine Population zu erzeugen, die  $n-1$  Kopien des besten Individuums der Ausgangspopulation enthält, wobei  $n$  die feste Populationsgröße ist.

#### Rangbasierte Selektion

Bei dieser Selektionsvariante steht die Selektionswahrscheinlichkeit  $p_s$  nicht mehr im direkten Verhältnis zur Fitness. Stattdessen werden die Chromosomen der Population absteigend nach ihrem Fitnesswert sortiert und mit Nummern versehen. Die Selektionswahrscheinlichkeit steht nun im Verhältnis mit der so entstandenen Rangzahl eines Chromosoms.

#### Wettkampfselektion

Dieser Selektionsalgorithmus ist zugleich auch ein Auswahlverfahren. Hierbei wird eine Menge von Individuen (mindestens 2, höchstens  $n$ ) aus der Population der Größe  $n$  ausgewählt, deren Fitness-Werte verglichen und das Beste in den mating pool kopiert. Dieses Vorgehen wird  $n$ -mal wiederholt, bis  $n$  Werte im mating pool vorhanden sind. Mit Hilfe der Anzahl der jeweils gezogenen Chromosomen kann der Selektionsdruck eingestellt werden — je mehr Individuen ausgewählt werden, desto größer ist der Selektionsdruck. Allerdings kann es bei diesem Verfahren passieren, dass im mating pool zum Schluss  $n$  identische Individuen liegen.

### 5.3.2 Auswahlalgorithmen

#### Roulette-Prinzip

Das Roulette-Prinzip lässt sich anschaulich durch ein Roulette-Rad darstellen, das in  $n$  Abschnitte unterteilt ist, wobei  $n$  wieder die feste Populationsgröße ist. Die Breite eines Abschnitts ist hierbei direkt proportional zur Selektionswahrscheinlichkeit eines Individuums. Nun wird  $n$ -mal eine Roulette-Kugel in das sich drehende Rad geworfen und das Individuum in den mating pool kopiert, in dessen Abschnitt die Kugel zur Ruhe gekommen ist. Je höher die Fitness eines Chromosoms ist, desto wahrscheinlicher landet die Kugel auch in seinem Abschnitt. Allerdings besteht auch hier die Gefahr, dass  $n$ -mal dasselbe Individuum ausgewählt wird.

#### Stochastic Universal Sampling (SUS)

Bei diesem Verfahren ist die Darstellung durch ein Glücksrad möglich, das in  $n$  proportional zur Selektionswahrscheinlichkeit eines Individuums große Abschnitte unterteilt ist. Außerdem werden  $n$  Zeiger im gleichmäßigen Abstand rund um das Rad angeordnet. Nun wird das Rad einmal gedreht, und von jedem Individuum werden so viele Kopien in den mating pool gelegt wie Zeiger auf den zugehörigen Abschnitt zeigen. Durch diese Vorgehensweise kann ausgeschlossen werden, dass  $n$  gleiche Individuen im mating pool landen.

## 5.4 Rekombination/Kreuzung

Der wichtigste genetische Operator bei genetischen Algorithmen ist der Kreuzungs-Operator (auch Crossover-Operator). Aus zwei Individuen der Elterngeneration werden Nachkommen erzeugt, indem Gene der Eltern kopiert und an die Kinder weitergegeben werden.

Die Auswahl der Eltern erfolgt stochastisch durch eines der bereits erwähnten Auswahlverfahren. Die Crossover-Wahrscheinlichkeit  $p_c$  gibt an, mit welcher Wahrscheinlichkeit überhaupt eine Kreuzung zwischen den beiden Eltern stattfindet. Es existieren verschiedene Kreuzungsverfahren, auf die im folgenden nun genauer eingegangen werden soll.

### 5.4.1 One-Point-Crossover

Durch eine Zufallszahl wird die Kreuzungsstelle ermittelt, an der die zwei Elternteile zerschnitten werden. Das erste der beiden erzeugten Nachkommen erhält den ersten Teil der Chromosomen des ersten Elternteils und den Teil ab der Kreuzungsstelle vom zweiten Chromosom. Der zweite erzeugte Nachkomme erhält die restlichen Gene, d. h. den ersten Teil des zweiten und den zweiten Teil des ersten Elternteils.

#### 5.4.2 N-Point-Crossover

Im Gegensatz zum One-Point-Crossover existieren hier mehrere Kreuzungsstellen. Das erste Kind erhält die Gene des ersten Elternteils bis zur ersten Kreuzungsstelle, anschließend bis zur nächsten Kreuzungsstelle die entsprechenden Gene des zweiten Elternteils. Nun werden wieder die Chromosomen des ersten Elternteils übernommen usw. Das zweite Kind erhält analog jeweils die Chromosomen des anderen Elternteils.

#### 5.4.3 Template-Crossover

Bei dieser Methode wird zunächst eine Schablone in der Länge der Chromosomen zufällig erzeugt. Die Schablone besteht wie die beiden Elternteile aus Einsen und Nullen. Das erste Kind erhält für jede 1 in der Schablone das entsprechende Allel vom ersten Elternteil sowie für jede 0 das vom zweiten. Beim zweiten Kind wird das Verhalten bei 0 und 1 vertauscht.

#### 5.4.4 Uniform Crossover

Beim Uniform Crossover wird für jedes Bit einzeln getestet, ob es zwischen den beiden Elternteilen ausgetauscht wird oder nicht. Hierfür maßgeblich ist eine festzulegende Wahrscheinlichkeit  $p_{ux}$  und eine bitbezogene Wahrscheinlichkeit  $U_z$  ( $z = 1, 2, \dots, n$ ). Falls  $p_{ux} \geq U_z$ , so werden die Bits zwischen den Elternteilen an der Position  $z$  vertauscht.

#### 5.4.5 Shuffle Crossover

Bei diesem Kreuzungsverfahren werden die Gene der Elternteile zunächst durchnummeriert und danach gemischt. Auf den durchgemischten Chromosomen wird nun ein One-Point- bzw. N-Point-Crossover durchgeführt. Zuletzt werden die Gene wieder entsprechend ihrer Nummerierung angeordnet.

### 5.5 Mutation

Unter einer Mutation versteht man das Flippen eines Bits, also den Übergang von einer 1 zu einer 0 und umgekehrt. Alternativ kann man die Mutation auch als zufällige Neubestimmung eines Bits betrachten. In diesem Fall kann es aber passieren, dass das mutierte Bit identisch mit seinem Vorgänger ist. Die Mutation soll vor allem verhindern, dass einzelne Allele bei allen Individuen der Population identisch sind, was bedeuten würde, dass die Suche nur noch in einem Unterraum des ursprünglichen Suchraums stattfinden würde.

Die Wahrscheinlichkeit, dass ein Bit mutiert, ist üblicherweise relativ gering. Mutationswahrscheinlichkeiten von  $p_m = 0,01$  und  $p_m = 0,001$  sind bei genetischen Algorithmen recht häufig.

## 5.6 Ersetzungsschema

Nachdem mit Hilfe der genannten genetischen Operatoren neue Nachkommen erzeugt wurden, muss entschieden werden, welcher Anteil der Eltern durch die Kinder ersetzt werden soll. Die Populationsgröße, die bei den Evolutionsstrategien mit  $\mu$  bezeichnet wird, muss am Ende des Iterationsschritts wieder so groß sein wie zuvor. Auch hierfür gibt es eine Reihe von Verfahren.

### 5.6.1 General Replacement

Hierbei werden alle Eltern durch die Kinder ersetzt. Dieses Verfahren läuft allerdings Gefahr, das beste Chromosom der Vorgängergeneration zu verlieren und dadurch die mittlere Güte der Gesamtpopulation zu verringern. Andererseits ist die Chance, sich auf wenige gute Individuen festzulegen (lokales Maximum), relativ gering.

### 5.6.2 Prinzip der Eliten

Falls eine Teilmenge der besten Individuen der Elterngeneration beibehalten wird, spricht man vom Prinzip der Eliten. Allerdings ist hierbei die Gefahr, sich auf ein lokales Maximum festzulegen, relativ hoch.

### 5.6.3 Schwacher Elitismus

Dieses Prinzip ähnelt dem vorherigen, jedoch werden hierbei die beizubehaltenden Chromosomen vor der Übernahme in die neue Generation mutiert.

### 5.6.4 delete-n-last-Schema

In diesem Verfahren werden die  $n$  schlechtesten generierten Kinder durch Individuen aus der Elterngeneration ersetzt.

## 5.7 Abbruchkriterium

Ein genetischer Algorithmus läuft in der Regel 50 bis mehrere hundert Generationen, bis das beste Individuum der letzten Generation als Lösung ausgegeben wird. Mögliche Abbruchkriterien sind beispielsweise das Erreichen einer bestimmten Güte oder eine gewisse Anzahl an Iterationen.

## 6 Anwendung eines genetischen Algorithmus

Als Beispielanwendung eines genetischen Algorithmus sei die Funktion  $f(x) = x^2$ ,  $0 \leq x \leq 31$  gegeben, deren Funktionswert nun maximiert werden soll.

Die **Kodierung** der Entscheidungsvariablen soll durch einen Bitstring der Länge 5 erfolgen.

Die **Ausgangspopulation** besteht aus vier Individuen und wird zufällig erzeugt.

Die **Bewertung** der Individuen erfolgt über ihre Funktionswerte. Folgende Tabelle veranschaulicht dies:

Nr.	Genotyp	Phänotyp	$f(x) = x^2$	$p_s$ in %	# Kopien im m.p.
1	01101	13	169	14.4	1
2	11000	24	576	49.2	2
3	01000	8	64	5.5	0
4	10011	19	361	30.9	1
$\Sigma$			1170	100.0	4

Die Selektionswahrscheinlichkeiten  $p_s$  errechnen sich durch  $\frac{f(x_i)}{\sum_i f(x_i)}$ .

Sie sind also proportional zum Anteil des Nutzenwerts von  $x_i$  am Gesamtnutzen (Prinzip der **fitnessproportionalen Selektion**).

Mit dem **Roulette-Prinzip** werden vier Kandidaten für den mating pool ausgewählt und in diesen kopiert.

Im mating pool befinden sich nun vier Individuen, die nach dem Zufallsprinzip miteinander gekreuzt werden. In unserem Beispiel sollen das zweite und das erste sowie das dritte und das vierte miteinander gekreuzt werden. Die gewählte Kreuzungsvariante ist hier das **One-Point-Crossover**.

Die Kreuzungsstelle ist ebenfalls per Zufall ermittelt worden. Die ersten beiden Chromosomen werden nach der 4. Stelle zerschnitten, während die anderen beiden an der 2. geteilt werden sollen.

Der mating pool besteht nun aus den folgenden Individuen:

Nr.	Genotyp	Phänotyp	$f(x) = x^2$	$p_s$ in %
1	01100	12	144	8.2
2	11001	25	625	35.6
3	11011	27	729	41.6
4	10000	16	256	14.6
Gesamt:			1754	100.0

Zuletzt unterziehen sich die Allele der einzelnen Chromosomen der **Mutation**. Da die Mutationswahrscheinlichkeit  $p_m$  üblicherweise sehr gering ist, nehmen wir hier den Wert  $p_m = \frac{1}{100}$  an.

In unserem Beispiel gibt es jeweils 20 Allele pro Generation, d. h. man kann pro Generation  $20 * \frac{1}{100} = 0.2$  Mutationen erwarten. Da dieser Wert immer noch sehr gering ist, mutiert in dieser Generation keines der 20 Bits.

Die neue Generation besteht nun aus vier neuen Individuen, deren Gesamtnutzen von 1170 auf 1754 gestiegen ist. Der durchschnittliche Nutzenwert ist ebenfalls von 293 auf 439 gestiegen. Iteriert man diesen Algorithmus über einige Generationen, wird die optimale Lösung 11111 bzw. 31 mit dem Gesamtnutzen 961 unter Umständen sehr schnell erreicht.

## 7 Zusammenfassung

Genetische Algorithmen sind eine Klasse von Optimierungsverfahren, die mit Hilfe von Operatoren der natürlichen Evolution eine Menge von Lösungen verwalten und in ihnen eine Suche nach einer optimalen Lösung durchführen. Sie eignen sich besonders gut für Probleme, deren genaue Struktur nicht bekannt ist („Black-Box-Optimierung“) oder die nicht in polynomieller Zeit gelöst werden können.

Da jedoch sehr viele Varianten von evolutionären Algorithmen existieren, ist es sicherlich eine Motivation für zukünftige Forschungsprojekte, weitere Anwendungsgebiete zu erschließen und neue Ideen für die Lösung von Optimierungsproblemen zu entwickeln.



## 8 Glossar

### **Allel**

die konkrete Ausprägung eines Gens, d. h. der Wert, den das Gen annehmen kann

### **Chromosom**

Bitstring, der eine kodierte Lösung repräsentiert

### **Crossover**

der Auswechselfvorgang von genetischem Material zwischen Chromosomen

### **Eltern**

zur Reproduktion ausgewählte Lösungen

### **Fitness**

Maß für die Reproduktionschancen eines Individuums

### **Gen**

Einheit eines Chromosoms; ein Bit des kodierten Bitstrings

### **Generation**

Iteration des genetischen Algorithmus

### **Genetik**

Lehre von der Ausbildung erblicher Merkmale und deren Vererbung in die weiteren Generationen

### **Genetische Algorithmen**

Spezielle Form der computergestützten Simulation der Evolution; begründet von John Holland Anfang der 60er Jahre

### **Genotyp**

kodierte Lösung von Entscheidungsvariablen

### **Individuum**

s. Chromosom

### **Kinder**

s. Nachkommen

### **Mutation**

Invertieren eines oder mehrerer Allele eines Chromosoms

### **Nachkommen**

aus den Eltern erzeugte Lösungen

### **Phänotyp**

dekodierte Lösung

**Population**

Menge von gleichartig strukturierten Individuen

**Rekombination**

Vereinigung von Genmaterial der Eltern zu neuen Individuen

**Selektion**

natürliche Auslese von Individuen

**Selektionsdruck**

äußere Einflüsse, die ein Individuum zur Anpassung zwingen

**Survival of the fittest**

engl.: das Überleben des Stärksten, des am besten Angepassten

## Literatur

- [1] Goldberg, David: Genetic algorithms in search, optimization, and machine learning; *Addison-Wesley-Verlag, 1989*
- [2] Holland, John: Adaptation in Natural and Artificial Systems; *The University of Michigan Press, 1975*
- [3] Nissen, Volker: Einführung in Evolutionäre Algorithmen; *Vieweg-Verlag, Braunschweig 1997*
- [4] Rechenberg, Ingo: Evolutionsstrategie. Optimierung technischer Systeme nach Prinzipien der biologischen Evolution; *Frommann-Holzboog-Verlag, Stuttgart 1973*
- [5] Schöneburg, Eberhard u. Heinzmann, Frank u. Feddersen, Sven: Genetische Algorithmen und Evolutionsstrategien; *Addison-Wesley-Verlag, Bonn 1995<sup>1</sup>*
- [6] Schwefel, Hans-Paul: Evolutionsstrategie und numerische Optimierung. Dissertation; *TU Berlin, 1975*
- [7] <http://www.educeth.ch/informatik/interaktiv/genalg/information.html>  
(14.02.2003)
- [8] <http://www.ki.informatik.uni-frankfurt.de/lehre/WS2002/KI/skript/KI-2GA.pdf> (08.11.2002)
- [9] [http://www.in.tu-clausthal.de/~gottlieb/wv/vortrag\\_081195.ps](http://www.in.tu-clausthal.de/~gottlieb/wv/vortrag_081195.ps)  
(08.11.1995)
- [10] [http://ima-www.informatik.uni-hamburg.de/Teaching/WS\\_2002/Bits\\_und\\_Zellen/03\\_ga.pdf](http://ima-www.informatik.uni-hamburg.de/Teaching/WS_2002/Bits_und_Zellen/03_ga.pdf)  
(02.05.2003)
- [11] <http://www.informatik.uni-ulm.de/ni/Lehre/SS02/Evosem/einfuehrung.pdf>  
(24.04.2002)
- [12] [http://www.informatik.uni-ulm.de/ni/Lehre/SS02/Evosem/PS\\_GAs.pdf](http://www.informatik.uni-ulm.de/ni/Lehre/SS02/Evosem/PS_GAs.pdf)  
(29.04.2002)