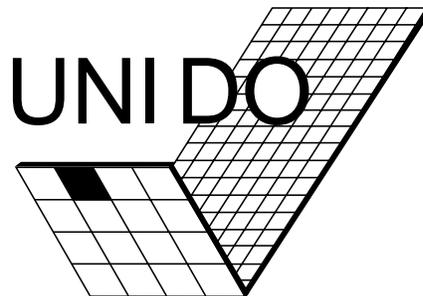


# Naturanaloge Optimierverfahren – Neuere Entwicklungen in der Informatik

Frank Kursawe

Fachbereich Informatik  
Lehrstuhl für Systemanalyse  
Postfach 50 05 00, D-4600 Dortmund 50



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Simulated Annealing</b>	<b>7</b>
<b>3</b>	<b>Neuronale Netze</b>	<b>11</b>
<b>4</b>	<b>Evolutionäre Algorithmen</b>	<b>19</b>
4.1	Evolutionsstrategien . . . . .	20
4.2	Genetische Algorithmen . . . . .	25
<b>5</b>	<b>Zusammenfassung / Ausblick</b>	<b>28</b>

## Zusammenfassung

In dieser Übersicht werden Problemlösungsverfahren vorgestellt, die Prinzipien aus der Natur ausnutzen, um mittels Computer-Hilfe möglichst das globale Optimum zu finden. Interessant sind dabei vor allem jene Bereiche der Natur (physikalische Abkühlprozesse, Zentralnervensysteme, biologische Evolution), in denen in einem „turbulenten Meer“ von Möglichkeiten gerade die „stabilen Inseln“ gefunden wurden. Die grundlegenden Ideen dieser Methoden sind — nach Maßstäben der Informatik — zum Teil schon recht alt, sie beginnen sich jedoch erst in jüngerer Zeit durchzusetzen. Es soll an dieser Stelle ein Überblick über die existierenden Ansätze gegeben werden, so daß Gemeinsamkeiten, Unterschiede und die Anwendbarkeit der abgeleiteten Algorithmen eingeschätzt werden können. Auf eine gesonderte Einführung in die Optimierung wird bewußt verzichtet und statt dessen an das Vorwissen der LeserIn appelliert.

# 1 Einleitung

In den letzten Jahren haben Algorithmen, die bestimmte Aspekte der Natur herausgreifen und imitieren, immer mehr Aufmerksamkeit auf sich gezogen. Dies hat einerseits damit zu tun, daß die enorm gewachsene Rechnerleistung die Simulation sehr viel komplexerer Modelle erlaubt, andererseits aber auch mit der Erkenntnis, daß man sich zur Beschreibung (Modellierung, Abbildung) bzw. Lösung realer Probleme am besten der Sprachen bzw. Strategien bedient, die das Urbild Natur selbst bereitstellt und seit Milliarden Jahren offensichtlich erfolgreich verwendet.

Die Übereinstimmung des Potentialgebirges der Mandelbrotmenge — als „Apfelmännchen“ bekannt geworden — mit den Bildern von schroffen Gletschern ist kein Zufall, sondern Ausdruck der universellen Gesetzmäßigkeit „Nichtlinearität“, die das Wechselspiel von Ordnung und Chaos regiert. Daraus ergeben sich einschneidende Konsequenzen für die Berechenbarkeit bzw. Vorhersagbarkeit jener Systeme, in denen nichtlineare Zusammenhänge auftreten: Obwohl streng deterministisch, kann die kleinste Veränderung der Anfangsbedingungen oder der kleinste Eingriff in einen laufenden Prozeß sich lawinenartig verstärken und den Ausgang völlig offen (chaotisch) erscheinen lassen. Das Bild vom Flügelschlag eines Schmetterlings, der — zumindest theoretisch — eine Wetterveränderung verursacht, dient oft als Veranschaulichung dieses Vorgangs. Kommen allerdings Rückkopplungs- und Ausleseprozesse hinzu, bietet sich dem System die Chance zur „Zähmung“ der Unberechenbarkeit, also zur *Selbstorganisation*: Der „Endzustand“ eines Systems kann zum Ausgangspunkt einer neuen Entwicklung werden; Störungen können abgefangen und für die eigene Weiterentwicklung genutzt werden.

Durch eine allgemeinere Beschreibungssprache wachsen die unterschiedlichsten Wissenschaftsdisziplinen enger zusammen: Das Verhalten physikalischer, chemischer, biologischer, medizinischer, soziologischer und wirtschaftlicher Prozesse läßt sich damit in einer gemeinsamen Sprache beschreiben. Das Ausbrechen von Panik hängt ebenso von der Zahl der spontan kopflos reagierenden Menschen ab wie die Durchsetzung des VHS- gegenüber dem Beta-Video-System von der anfänglich nur geringen Überzahl für VHS [5]. Vielleicht sind wir Zeugen eines Paradigmenwechsels: Das klassische Schubladendenken der Natur- und Geisteswissenschaften könnte abgelöst werden durch eine „Systemwissenschaft“, die in der materiellen Welt nach Bedingungen für die Bildung von Strukturen sucht. Nach der Erschütterung durch die Ideen der Quantenmechanik droht nun die endgültige Ablösung

des mechanistischen durch ein nichtlineares Weltbild. Dieser Vorgang könnte allerdings große Lücken an jenen Stellen hinterlassen, an denen insbesondere die Naturwissenschaften als Ersatzreligion aufgebaut wurden. Denn im Zweifel wissen wir bei Eingriffen in natürliche Systeme, die Nichtlinearitäten aufweisen, eben *nicht*, was wir tun, so daß auch die besten Absichten fatale Folgen haben können.

Deterministisches Chaos in Gehirnen scheint auch der Schlüssel zum Verständnis der Vorgänge „Lernen“ und „Erinnern“ zu sein: Ständig vorhandenes, aber räumlich begrenztes deterministisches Chaos erlaubt dem Gehirn, diesen Zustand nur durch Änderung der Randbedingungen (Sinnesreize) in Richtung auf einen Attraktor (Erlerntes) zu verlassen, während die Alternativerklärung „latentes Grundrauschen“ ein Muster nur durch langsames Abkühlen stabilisieren könnte. Andererseits kann ein Attraktor auch ebenso schnell wieder verlassen werden, um den Organismus erneut bereit für neue Sinnesreize werden zu lassen. Chaotisches Verhalten findet sich auch in der Herzfrequenz, in den Schwankungen der Menge weißer Blutkörperchen, bei Stoffwechselprozessen und Hormonkonzentrationen.

Neben den im folgenden ausführlicher vorgestellten naturanalogen Modellansätzen sollen zwei weitere erwähnt, aber in diesem Rahmen nicht detaillierter vorgestellt werden:

- *Zelluläre Automaten* gehen auf eine Idee John von Neumanns Anfang der fünfziger Jahre zurück, der auf der Suche nach einer formalen Beschreibung des Phänomens der Selbstreproduktion folgenden zellulären Automaten definierte [31]: Ein zweidimensionaler Automat mit 29 Zuständen und einer sogenannten Von-Neumann-Nachbarschaft (die Zelle selbst und ihre vier orthogonalen Nachbarn) ist einerseits so berechnungsuniversell wie eine Turing-Maschine und kann andererseits jeden anderen vorgegebenen Automaten konstruieren, insbesondere also auch sich selbst. Sogar das allseits bekannte „Game of Life“ mit seinen zwei Zuständen kann als berechnungs- oder Turing-universell angesehen werden [8].

In zellulären Automaten gibt es keine Trennung von Daten und Anweisungen, sondern nur Konfigurationen. Aus wenigen einfachen lokalen Regeln und Zuständen können — angetrieben von einer imaginären Uhr — vielfältige komplexe globale Strukturen entstehen. So läßt sich mit diesen Automaten zum Beispiel die Ausbreitung von Infektionskrankheiten simulieren. Ändert man für denselben Automaten nur die Semantik, so daß „krank“ nun „weiß“ und „gesund“ „schwarz“ bedeutet,

so beschreibt der Automat periodisch ablaufende chemische Reaktionen (z.B. die Belousov–Zhabotinskii–Reaktion).

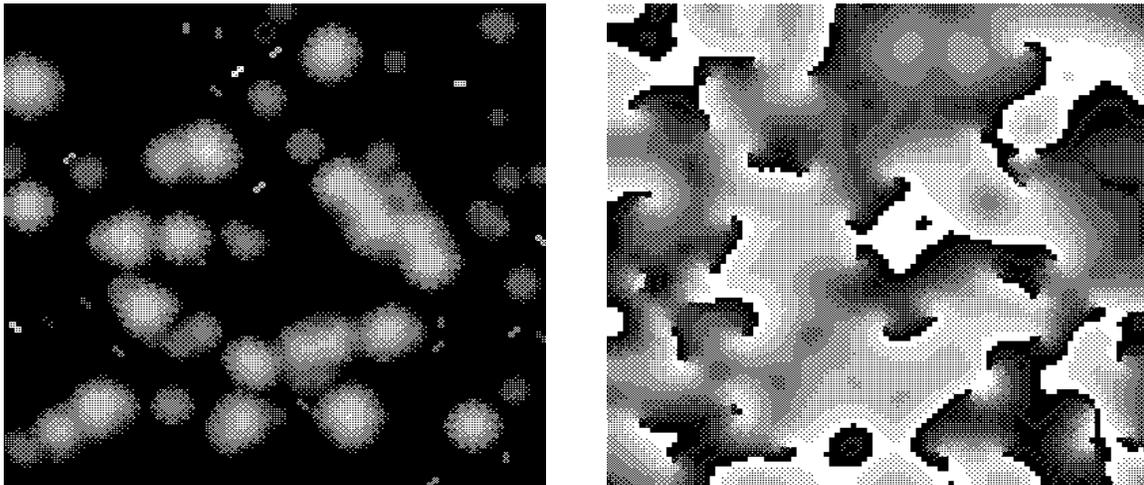


Abbildung 1: Ausbreitung einer Infektion nach 5 bzw. 70 Zeitschritten

Ferner gibt es Anwendungen, die mittels Regeln für lokale Partikelkollisionen das Verhalten von Gasen und Flüssigkeiten beschreiben. Und auch sogenanntes dendritisches, d.h. baumähnliches Wachstum wie etwa bei Schneeflocken oder Korallen läßt sich mit diesen Automaten simulieren [7].

Als Optimierverfahren erscheinen zelluläre Automaten allerdings ungeeignet, weshalb sie hier nicht weiter betrachtet werden.

- Modelle des *Immunsystems* zur Problemlösung einzusetzen, erscheint naheliegend, weil sich hier eine erstaunliche Anpassungsfähigkeit an eine sich ständig und nicht vorhersagbar ändernde Umwelt zeigt, sowohl was das Lernen neuer (unbekannte Eindringlinge) als auch das Wiedererkennen bereits aufgetretener Situationen angeht (Impfung). Im Immunsystem des Menschen produziert ein sich ständig veränderndes Netzwerk wahllos Unmengen von Antikörpern. Ein auf einen eingedrungenen Krankheitserreger „passender“ Antikörper ist für die ihn produzierende Zelle das Signal, sich zu vermehren, wodurch große Mengen dieses Antikörpers ausgeschüttet werden. Als etwas unbeholfene Veranschaulichung für den Informationsaustausch kann ein neuronales Netz (siehe Kapitel 3) dienen, dessen Struktur sich mit der Zeit verändert („idiotypisches Netzwerkmodell“).

Aussagen über die genaue Funktionsweise des Immunsystems sind derzeit noch im Hypothesen–Stadium, was für die Gültigkeit der entsprechenden Rechnersimulationen und ihrer Resultate entsprechend gilt. Dieser Ansatz ist also noch zu neu, um bereits in seiner Anwendbarkeit über die reine Simulation hinaus eingeordnet werden zu können; erste Resultate deuten auch eher auf einen driftenden Suchprozess als auf eine Optimierung hin. Daher wird auch diese Richtung hier ausgeblendet.

## 2 Simulated Annealing

Das Einfrieren und Auskristallisieren einer Flüssigkeit bzw. allgemein das Abkühlen (engl.: to anneal) einer Schmelze zum Kristall bildet das natürliche Vorbild für die Methode des *Simulated Annealing*. Bei hohen Temperaturen können sich die Moleküle noch relativ frei und unabhängig voneinander bewegen, doch diese Fähigkeit verliert sich im Zuge des langsamen Abkühlens mehr und mehr. Gibt man diesem Prozeß genug Zeit, wird eine Kristallstruktur auf minimalem Energieniveau erreicht. Kühlt man zu schnell ab, wird dieser für das System optimale Zustand nicht erreicht, und man erhält einen polykristallinen oder amorphen Körper höherer Energie. Die Spiegel der klassischen Teleskope, die nur aus einem großen Glaskörper bestehen, werden beispielsweise über viele Monate hinweg abgekühlt, damit später die Beobachtungen möglichst nicht durch Spannungen in der Spiegelstruktur verfälscht werden.

Auch wenn der Vergleich etwas hinkt, zeichnen sich viele der traditionellen Optimierverfahren durch schnelles „Abkühlen“ aus: Vom Startpunkt bewegt man sich solange auf- bzw. abwärts, wie man Erfolg hat. Daß auf diese Weise oft nur das nächstgelegene lokale Optimum gefunden werden kann, dürfte einsichtig sein.

Die Simulation natürlicher Kühlprozesse basiert dagegen auf der sogenannten *Boltzmann-Wahrscheinlichkeitsverteilung*

$$Prob(E) \sim \exp\left(\frac{-E}{kT}\right),$$

die anschaulich besagt, daß der energetische Zustand eines Systems im thermischen Gleichgewicht (bei Temperatur  $T$ ) einer Zufallsverteilung über allen möglichen Energiezuständen  $E$  unterliegt ( $k$  bezeichnet die Boltzmann-Konstante, die Energie mit Temperatur in Relation setzt). So kann ein System mit niedriger Temperatur durchaus in einem energetisch hohen Zustand sein, wenn auch die Wahrscheinlichkeit dafür exponentiell kleiner wird. Dadurch bietet sich die Chance, ausgehend von einem lokalen Optimum ein besseres zu finden, insbesondere auch das globale [30, 1]. Der Abkühlungsverlauf wird also keine monoton fallende Funktion sein; erst durch das Zulassen von zwischenzeitlichen Verschlechterungen erhöht man die Chance, ein besseres Resultat zu finden als nur das dem Startpunkt nächstgelegene lokale Optimum (siehe Abbildung 2 und 3).

Die Grundidee dieses Verfahrens stammt bereits aus dem Jahr 1953. Metropolis et al. wendeten die oben beschriebene Idee in der Thermodynamik an: Ein System wechselt von Energieniveau  $E_1$  nach Energieniveau  $E_2$  mit der Wahrscheinlichkeit

$$p = \exp\left(\frac{-(E_2 - E_1)}{kT}\right).$$

Für  $E_2 < E_1$  wird die Wahrscheinlichkeit bei  $p = 1$  „abgeschnitten“. Anschaulich bedeutet dies, daß eine Modifikation, die in einem geringeren Energiezustand resultiert, immer akzeptiert wird. Um diesen Algorithmus nicht nur im Kontext thermodynamischer Modelle zu verwenden, müssen folgende Elemente vorhanden sein [17]:

- eine geeignete Beschreibung der Systemzustände,
- ein Mechanismus, der zufällige Änderungen in diesen Zuständen vornimmt und dann dem System „anbietet“,
- eine Zielfunktion  $E$ , die minimiert werden soll, sowie
- ein Kontrollparameter  $T$  — das Analogon zur Temperatur — und ein Schema, das angibt, nach wievielen Zustandsänderungen  $T$  um welchen Betrag verringert wird.

Als konkretes Beispiel sei hier das *Problem des Handlungsreisenden* vorgestellt, das darin besteht,  $n$  Städte auf einer möglichst kurzen Rundreise genau einmal zu besuchen und zum Ausgangspunkt zurückzukehren. Dieses Problem ist auf der einen Seite ein Vertreter der Klasse sogenannter *NP-vollständiger* Probleme, die sich dadurch auszeichnet, daß die Rechenzeit für eine exakte Lösung exponentiell in  $n$  — der Problemgröße (hier: Zahl der Städte) — wächst. Das Ausprobieren aller möglichen geschlossenen Wege benötigt bereits für 30 Städte auf einem Rechner, der 1000 Wege in der Sekunde erzeugt und prüft, mehr Zeit als das Universum alt ist. Auf der anderen Seite weist es eine *multimodale* Zielfunktion auf, d.h. eine Funktion, die mehrere lokale Optima hat (siehe auch Abbildung 8 rechts). Wie sieht nun die genaue Problemformulierung aus?

- Die Städte sind von 1 bis  $n$  durchnummeriert und liegen auf den Koordinaten  $(x_i, y_i)$ . Mögliche Systemzustände sind dann gerade Permutationen der Reihenfolge, in der die Städte besucht werden.

- Zwei zufällige Änderungen einer Rundreise sind vorgesehen:
  - Ein Stück der Rundreise wird herausgetrennt und an gleicher Stelle in umgekehrter Reihenfolge wieder eingesetzt.
  - Ein Stück der Rundreise wird herausgetrennt und an einer anderen, zufällig ausgewählten Stelle wieder eingesetzt.
- Die Zielfunktion besteht einfach aus der Länge der Reise:

$$E = \sum_{i=1}^n \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \quad \text{mit } n + 1 := 1$$

- Ein gutes Abkühlschema zu finden, erfordert einige Experimente. Man wählt die Anfangstemperatur so, daß sie signifikant größer als das größte  $\Delta E$  ist, das man experimentell beobachtet hat. Befinden sich die Städte — wie in Abbildung 2 und 3 — in einem Einheitsquadrat, wäre z.B. 0.5 ein sinnvoller Startwert für die Temperatur  $T$ , die auf jedem Niveau solange unverändert bleibt, bis man z.B. entweder  $100n$  Änderungen vorgeschlagen oder  $10n$  bessere Konfigurationen, d.h. kürzere Rundreisen gefunden hat. Das Abkühlen kann im einfachsten Fall aus der Multiplikation der Temperatur mit einem festen, über der Zeit konstanten Faktor ( $< 1.0$ ) bestehen. Man kann sich aber auch Heuristiken überlegen, die diesen Faktor vom bisherigen Erfolg des Verfahrens abhängig machen. Treten auf einer Temperaturstufe keine Verbesserungen mehr auf, wird die Suche beendet.

Die Zielfunktion läßt sich ohne Aufwand gegen eine andere austauschen. Schwieriger dürften sich jeweils die Festlegung der Zufallskomponente und die Wahl des Abkühlungsschemas gestalten.

Die beiden folgenden Abbildungen zeigen die jeweils erreichte Rundreise und den Verlauf des Abkühlungsprozesses für  $n = 100$ . Im ersten Fall lieferte der Standard-Algorithmus die Resultate, im zweiten eine Variante, die keine Verschlechterungen akzeptiert.

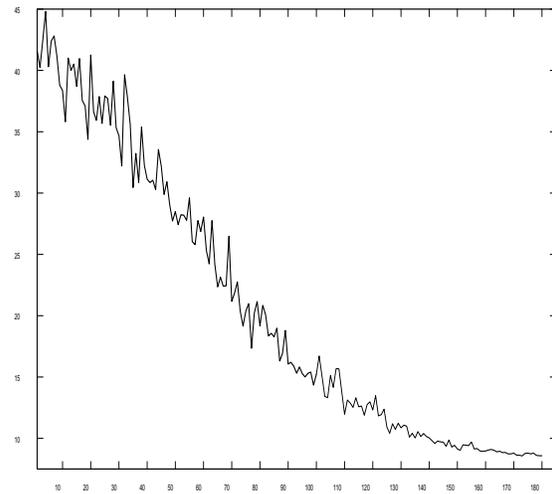
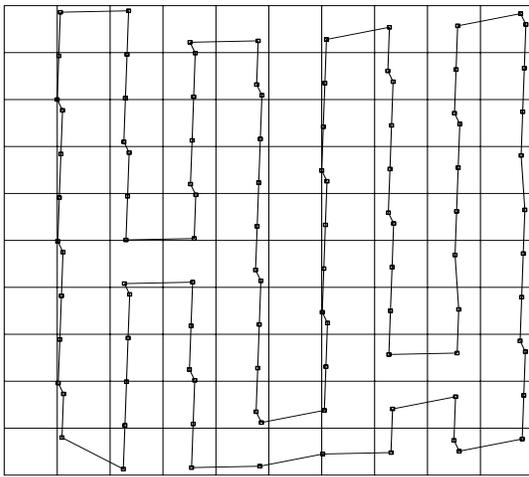


Abbildung 2: Rundreise: 8.5, Abkühlungsverlauf mit Verschlechterungen

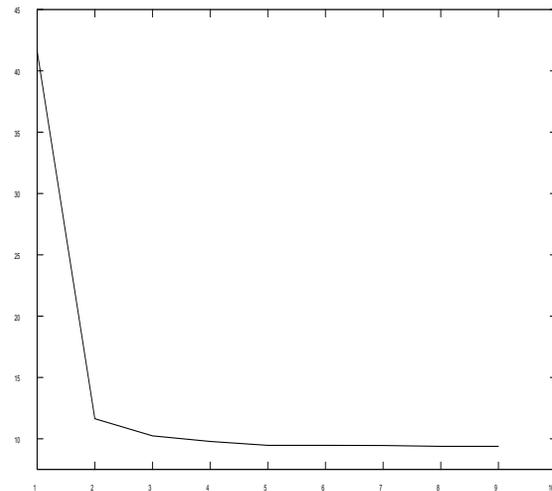
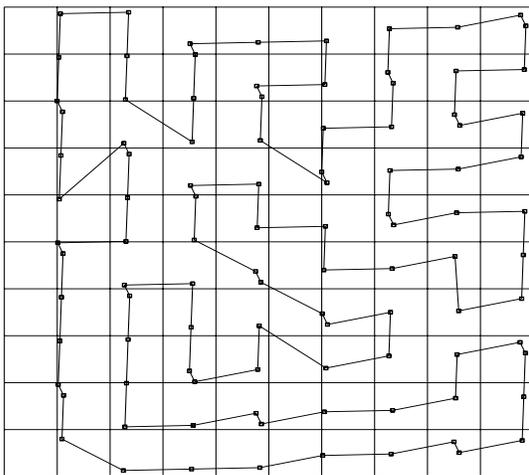


Abbildung 3: Rundreise: 9.4, Abkühlungsverlauf ohne Verschlechterungen

### 3 Neuronale Netze

Unter diesem Begriff werden jene Arten von Informationsverarbeitung zusammengefaßt, die in Anlehnung an die Funktionsweise von Nervensystemen auf dem Zusammenwirken vieler, parallel arbeitender *Neuronen* basieren, die über Synapsen und Axons miteinander kommunizieren können. So kann jedes Neuron im menschlichen Gehirn die Signale von bis zu zehntausend anderen Neuronen empfangen, und selbst zum entferntesten gibt es eine Verbindung über nur etwa zehn Zwischenstationen. Die Arbeitsweise kann man sich prinzipiell so vorstellen: Ein Neuron registriert die ankommenden Signale, bildet daraus eine gewichtete Summe und leitet einen aus ihr berechneten Wert an andere Neuronen weiter:

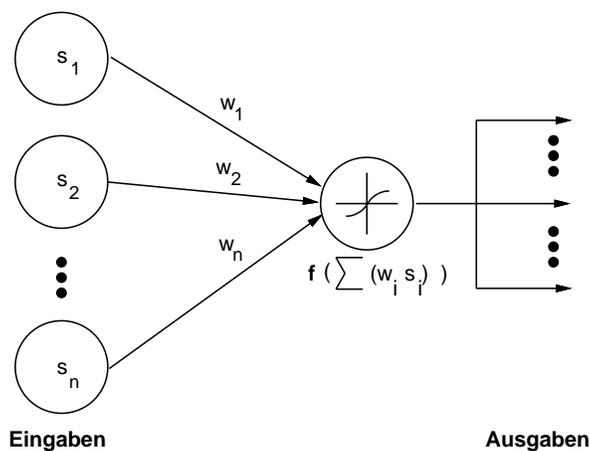


Abbildung 4: Arbeitsweise eines künstlichen Neurons

Das (kindliche) Lernen besteht darin, viele Synapsen absterben zu lassen und nur jene Verbindungen zu verstärken, die durch ein Muster aktiviert werden. Dieses einfache Modell — von dem Psychologen Hebb bereits 1949 vorgeschlagen [11] — mag ausreichen, um den Spieltrieb der Informatiker zu wecken, die genauen Vorgänge im Gehirn liegen allerdings noch ziemlich im Dunkeln. So soll nicht verschwiegen werden, daß dort die Kommunikation mit Hilfe von mindestens dreißig Botenstoffen vollzogen wird und Neuronen nicht nur eines Typs, sondern unterschiedlichster Struktur auftreten.

Neuronale Netze bilden einen Rahmen für das Lernen anhand von Erfolgen und Mißerfolgen. Obwohl sie Aufgabenbereiche mit der Künstlichen Intelligenz teilen (Mustererken-

nung und –vervollständigung, Klassifikation von Merkmalen, Assoziative Speicher, fehler-tolerierende Systeme, Sprachverarbeitung), unterscheiden sich diese Netze grundsätzlich von der herkömmlichen Wissensverarbeitung und –repräsentation: Das Wissen ist nicht mehr wie etwa in einzelnen Regeln eines Expertensystems lokalisierbar, sondern eher nach Art eines Hologramms über die ganze Struktur verteilt.

Bereits 1958 hatte Rosenblatt das „Perceptron“ entwickelt [24], dessen Eigenschaften bis dahin einmalig waren: Dieses einfache Modell war lernfähig und fehlertolerierend, konnte Muster klassifizieren und zeigte dabei sogar eine gewisse Abstraktionsfähigkeit, indem es auf unbekannte Eingaben korrekt antwortete. Es konnte ferner gezeigt werden, daß das Lernverfahren nach endlich vielen Schritten alle Muster klassifiziert, zu denen es prinzipiell fähig ist. Zu einem Rückschlag kam es jedoch, als 1969 Minsky und Paperts Kritik an den Schwächen der Perceptron–Netzwerke — Notwendigkeit des Lernens der Synapsengewichte mit exponentiell mit der Problemkomplexität wachsender Genauigkeit — fälschlicherweise auch auf Netzwerke mit mehr als zwei Schichten übertragen wurde [20]. Erst in den 80er Jahren hat sich dieses Gebiet durch die Arbeiten von Hopfield [15] sowie Rumelhart und Hinton [26] davon erholen können.

Die wichtigsten Bestandteile eines neuronalen Netzes sind

- die Verarbeitungselemente (Neuronen) mit
  - einer Menge von Aktivierungszuständen,
  - einer Aktivierungsfunktion,
  - einer Ausgabefunktion, abhängig vom aktuellen Zustand,
- eine Verbindungstopologie zwischen ihnen und
- eine Lernregel, um die Synapsengewichte anzupassen.

Eine bestimmte Wahl der oben genannten Elemente realisiert einen bestimmten Netzwerktyp. Heute lassen sich folgende wichtige Modelle neuronaler Netze unterscheiden [21, 23]:

- Einem *Perceptron*, bestehend aus einer Schicht mit einer festen Anzahl von Neuronen, werden über Leitungen Muster als Eingaben zugeführt. Anhand einer endlichen Anzahl eingegebener Muster, die jeweils  $m$  Merkmale umfassen, sollen diese korrekt

in Klassen eingeordnet werden. Daß ein gegebenes Netz dieser Art eine bestimmte Aufgabe überhaupt lösen kann, ist nicht a priori gesichert. Die Muster müssen durch eine  $(m - 1)$ -dimensionale Hyperebene separierbar sein. Ineinander verschlungene Spiralen lassen sich nicht durch eine Gerade trennen (siehe Abbildung 5), folglich hat das Perceptron keinerlei Aussicht auf Erfolg. Diese Situation läßt sich aber oft durch Hinzunahme eines weiteren Merkmals entschärfen.

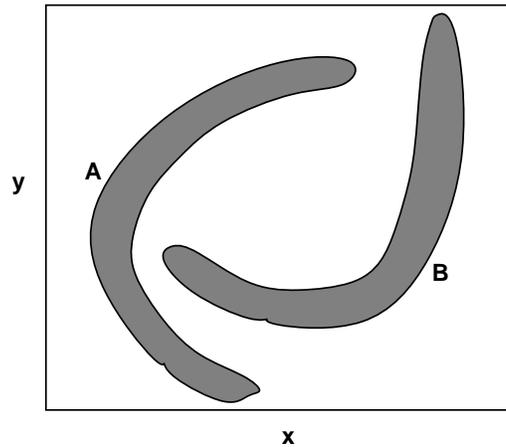


Abbildung 5: In der  $(x, y)$ -Ebene nicht linear separierbare Muster  $A, B$

Auch an einem so einfachen Problem wie dem, das logische „exklusive Oder“ zweier binärer Eingaben zu berechnen, scheitert dieses Netz, da es nicht möglich ist, die vier Punkte  $(0,0)$ ,  $(1,1)$ ,  $(0,1)$  und  $(1,0)$ , von denen die ersten beiden die Klasse 0 und die letzten beiden die Klasse 1 bilden, durch eine Gerade zu trennen.

Der limitierte Einsatzbereich des Perceptrons wird neuerdings durch mehrlagige Netze zumindest teilweise überwunden:

- Das mehrschichtige *Perceptron mit Backpropagation* dürfte das bekannteste Modell neuronaler Netzwerke sein. Ein solches Netz zu trainieren, ist allerdings erst möglich, seit es Rumelhart, Hinton und Williams [26] gelang, die sogenannte Delta-Regel, deren Anwendbarkeit sich auf Netze mit nur je einer Ein- und Ausgabeschicht beschränkte, zum rekursiven Backpropagation-Lernverfahren weiterzuentwickeln. Mit diesem Modell lassen sich fast beliebige Abbildungen leisten, da die zusätzlichen, verborgenen Schichten die durchlaufenden Informationen kodieren und komprimieren. Abbildung 6 zeigt die Topologie eines solchen Netzes:

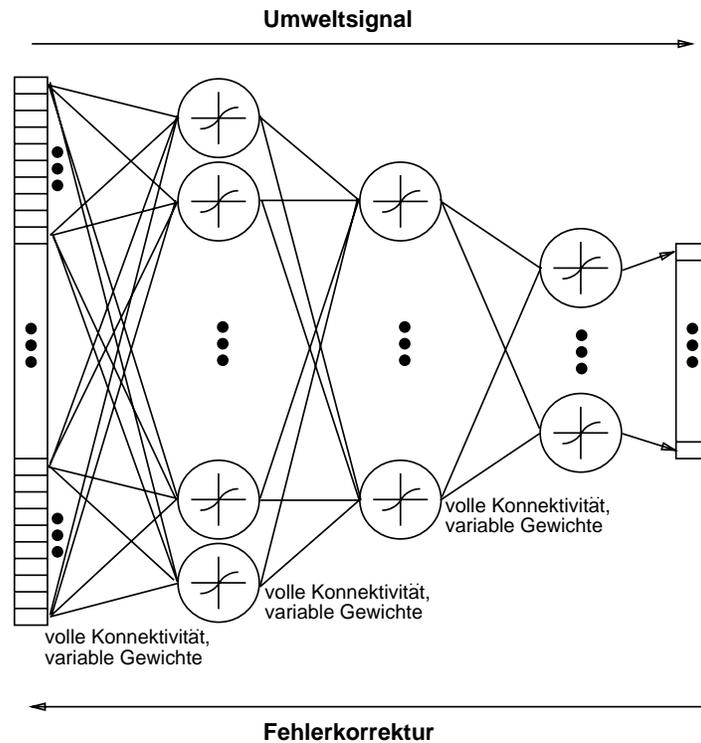


Abbildung 6: Ein dreischichtiges Backpropagation-Netz

Folgende „sigmoide“ Funktion wird häufig benutzt, um die Antwort eines Neurons auf seinen gesamten Input zu definieren:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Der Backpropagation-Algorithmus versucht, ausgehend von zufällig eingestellten Startwerten die Synapsengewichte in kleinen Schritten so einzustellen, daß der Fehler zwischen Eingabe und erwarteter Ausgabe minimal wird. Bei geeignet gewählter Schrittweite (Lernrate) bewegt man sich in Richtung des steilsten Abstiegs (Gradienten) der Fehlerfunktion. Unter Verwendung der oben angegebenen Sigmoid-Funktion ergibt sich als Korrekturterm für die Gewichte der Ausgangsschicht

$$\Delta w_{jk} = \eta o_j \delta_k,$$

mit:

- $\Delta w_{jk}$ : Gewicht zwischen Neuron  $j$  der vorgeschalteten und Neuron  $k$  der Ausgangeschicht,
- $\eta$ : Lernrate aus dem Intervall  $[0, 1]$ ,
- $o_j$ : Ausgabewert des Neuron  $j$  der vorgeschalteten Schicht,
- $\delta_k = o_k (1 - o_k) (t_k - o_k)$  mit:
  - \*  $o_k$ : berechneter Ausgangswert von Neuron  $k$  der Ausgangeschicht und
  - \*  $t_k$ : gewünschter Ausgangswert von Neuron  $k$  der Ausgangeschicht.

Für die Korrektur in den verborgenen Schichten verändert sich die Formel für  $\delta_k$  etwas, da man den Fehler über alle nachgeschalteten Ebenen summieren muß. Für eine detailliertere Darstellung sei auf die Literatur verwiesen.

Dieses Verfahren ist auf Netze mit beliebig vielen internen Schichten anwendbar. Der Trainingsaufwand für solche Netze kann allerdings beträchtlich werden, und das Lernverfahren kann in lokalen Minima steckenbleiben. Trainiert man ein Netz nur mit idealen Mustern, wird dieses Netz schon bei einer geringen Störung Schwierigkeiten mit der Entscheidung haben. Ferner ist ungeklärt, wieviele verborgene Schichten mit welcher Neuronenzahl man für ein gegebenes Problem benötigt und was passiert, wenn man das Netz überdimensioniert.

- *Hopfield-Netze* weisen viele Gemeinsamkeiten mit dem in Kapitel 2 vorgestellten Verfahren des Simulated Annealing auf. Hopfield konnte 1982 [15] zeigen, daß das Verhalten der nach ihm benannten Klasse von Netzwerken dem einer großen Anzahl von Elementarmagneten („Spingläser“) entspricht und sich dies in einer gemeinsamen Energiefunktion ausdrückt, wodurch Methoden aus der Physik zur Analyse solcher Netze verwendet werden konnten. So läßt sich der maximal in einem Hopfield-Netz mit  $n$  Neuronen speicherbare Informationsgehalt analytisch zu  $2n$  berechnen. Diese Zahl verringert sich allerdings, falls auch noch mit Rauschen überlagerte Muster erkannt werden sollen.

Anschaulich ausgedrückt, rollt das Modell eine Kugel durch das Zielfunktionsgebirge der „Energiefunktion“ und sucht nach einer möglichst tief liegenden Mulde. Die Mulden entsprechen gerade den erlernten Mustern. Ihre räumliche Ausdehnung und

die Monotonie des Weges dorthin legt fest, wie anfällig (oder auch nicht) der Lauf der Kugel gegen Störungen der Eingänge ist.

Alle hier vorgestellten Netzwerkmodelle basieren auf den oben erwähnten Grundbausteinen. Die neuartige Charakteristik der Hopfield–Netze liegt in der Rückkopplung: Die Ausgänge der einschichtigen Topologie werden an die Eingänge mit der Zielsetzung zurückgeführt, ggf. fehlende oder verrauschte Teile schrittweise zu ergänzen. Ein herkömmlich programmierter Rechner muß mehr oder weniger viele gespeicherte Muster mit dem eingegebenen vergleichen und kann dann das am wenigsten abweichende ausgeben, während ein gut angelernter *assoziativer Speicher* seine Ausgabe nach abstrakten, holistischen Ähnlichkeitskriterien berechnet. Als assoziativer Speicher ist dieser Netzwerktyp jedoch nur bedingt geeignet, da seine Abstraktionsfähigkeit nur schwach entwickelt ist. Gerade in diesem Bereich liegt aber die Leistungsfähigkeit der „Urbilder“ — echter Gehirne — verborgen, da die Umwelt für uns selten Informationen bereithält, die perfekt auf etwas Gelerntes passen.

- *Kohonen Feature–Maps* sind als zweidimensionale Neuronenschichten organisiert [18]. Die eingehenden Signale rufen bei diesem Modell im Gegensatz zum Backpropagation–Verfahren nur eine räumlich begrenzte Aktivität hervor, und zwar derart, daß ähnliche Reize nur eine Erregung benachbarter Neuronen verursachen, wie dies in der Natur zum Beispiel auf der Retina beobachtet wird. Das Lernen geht so vor sich, daß ausgehend von zufällig voreingestellten Synapsenstärken dem Netzwerk ebenfalls stochastisch ausgewählte Stimuli (Vektoren) präsentiert werden. Das Neuron, das sich dadurch am stärksten „angesprochen“ fühlt, bewegt sich dann ein wenig auf den Vektor zu und zieht seine Nachbarn in einem räumlich begrenzten Gebiet, das mit zunehmender Lerndauer immer kleiner wird, mit. Die Geschwindigkeit, mit der die Nachbarschaften kleiner werden, hat erheblichen Einfluß auf die Resultate: Bei zu rascher Verkleinerung der Nachbarschaften gerät das Netz unter „Streß“ und „friert“ in einem Zustand ein, der nicht optimal ist; andernfalls verschenkt man Rechenzeit.

Auch hier soll als konkrete Anwendung wieder das Handlungsreisenden–Problem betrachtet werden: Ein Neuronennetz soll eine möglichst kurze Rundreise durch  $n$  Städte lernen. Pro Iteration wird eine Stadt  $S_{x,y}$  zufällig ausgewählt und zieht das ihr nächstgelegene Netzelement  $N_{x,y}$  etwas zu sich, deformiert also das Netz etwas. Eine bestimmte Anzahl der Nachbarn macht diesen Schritt mit. Je weiter diese Nachbarn

von  $N_{x,y}$  entfernt sind, umso geringer fällt ihre Kooperationsbereitschaft aus. Mit zunehmender Dauer nimmt diese Kooperation immer weiter ab, und man kann beobachten, wie sich nach Ausbildung einer Grobstruktur die Neuronen immer mehr an die Städtepositionen annähern, bis sie schließlich mit ihnen übereinstimmen.

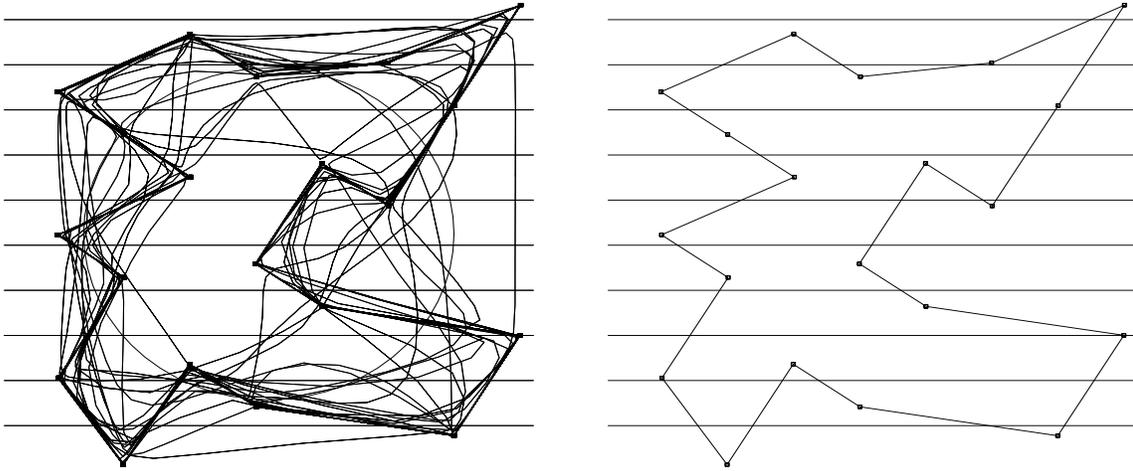


Abbildung 7: Entwicklung einer Rundreise, optimale Rundreise

Abbildung 7 zeigt die Entwicklung eines solchen 60-elementigen Netzes (links) — anfänglich eine Kreisform — und das stabilisierte Muster der optimalen Rundreise (rechts) durch 20 Städte, die nur etwa 10 Sekunden dauert. Nur aus Gründen der Übersichtlichkeit wurde eine so kleine Anzahl von Städten gewählt.

Erweiterungen dieses Modells sind für die Robotersteuerung attraktiv, da dieses Verfahren nicht nur Nachbarschaften erhalten bzw. lernen kann, sondern auch Regionen, in die häufiger eine Reizung fällt, mit entsprechend mehr Neuronen ausstatten kann als seltener angesprochene Gebiete. Als biologisches Vorbild für diesen Vorgang dient z.B. die Retina, auf der die Nervenzellen sich um den Sehnerv besonders konzentrieren. Simulationen leisten bereits die Bewegungssteuerung eines Roboters und die vertikale Balancierung eines Stabes.

- *Carpenter/Grossberg-Classifer* kommen der wirklichen Funktion des Gehirns wohl am nächsten, da sie mittels Rückkopplungen, Lang- und Kurzzeitgedächtnis sowie einer Art Aufmerksamkeitsfunktion, die ein Absinken der Aktivität bewirkt, falls keine Reize vorliegen, erstmals simulieren können, wie das Gehirn dazulernt, ohne

dabei bereits Gelerntes wieder zu zerstören, wie es also einerseits Inseln der Stabilität, andererseits aber auch Regionen mit chaotisch anmutender Aktivität gibt. Die Grundlage für diesen Ansatz bildet die von Grossberg in den 70er Jahren entwickelte „Adaptive Resonance Theory“ (ART). An dieser Stelle auf Details einzugehen, würde allerdings den Rahmen sprengen.

Die folgende Tabelle faßt die wesentlichen Eigenschaften der hier vorgestellten Netzwerktypen noch einmal zusammen [3]:

<i>Netzwerkmodell</i>	<i>Topologie</i>	<i>Werte</i>	<i>Lernen</i>
Perceptron	eine Schicht vorwärts verbunden	binär und kontinuierlich	beaufsichtigt
Perceptron mit Backpropagation	mehrere Schichten vorwärts verbunden	binär und kontinuierlich	beaufsichtigt
Hopfield	eine Schicht mit Rückkopplung	binär	beaufsichtigt
Kohonen Feature Maps	2-dimensionales Raster von Ausgabeelementen	kontinuierlich	unbeaufsichtigt
Carpenter/Grossberg- Classifier	mehrere Schichten	binär und kontinuierlich	unbeaufsichtigt

Endgültig wird sich die Anwendungsbreite neuronaler Netze aber erst einschätzen lassen, wenn Hardware ohne feste Verdrahtung, d.h. optische Chips, zur Verfügung steht, wenn also der Verbindungsaufbau dynamisch erfolgen kann und somit die Netztopologie nicht mehr von Programmen auf sequentiellen Rechnern simuliert werden muß. Allerdings kann man von einer reinen Vermehrung der Neuronenzahl keinen beliebigen Leistungszuwachs erwarten, denn dann geht ein solches Netz den bequemen Weg und „merkt“ sich die Eingabemuster, statt sie zu lernen.

## 4 Evolutionäre Algorithmen

Da die beiden hier vorgestellten Verfahren (Evolutionsstrategien, Genetische Algorithmen) sich relativ ähnlich sind und auf der Nachahmung biologischer Evolutionsprinzipien beruhen, erfolgt eine gemeinsame Einführung.

Die Idee, Prinzipien der biologischen Evolution unter Optimierungsgesichtspunkten zu interpretieren, wurde — trotz einiger auch technisch bemerkenswerter Resultate in der Natur — lange Zeit belächelt oder sogar rigoros abgelehnt, weil

- gemessen an der Lebenszeit eines Menschen die Evolution sehr langsam und nicht direkt beobachtbar verläuft,
- keine globale „Zielfunktion“ identifizierbar ist,
- ein auf stochastisch beschriebenen Störungen basierendes Mutations–Selektions–Verfahren aufwendig und verschwenderisch gegenüber den für einen Spezialfall entwickelten Strategien wirkt,
- bei ersten Simulationsversuchen der Slogan „Survival of the Fittest“ zu wörtlich genommen wurde,
- das der Evolution inhärente Zufallselement oft falsch interpretiert wird: Einfaches „Würfeln“ kann in der Tat die heutige Vielfalt an komplexen Lebewesen nicht hervorgebracht haben; entsprechend erfolglos muß das gleichverteilte Raten im Parameterraum einer Zielfunktion sein. Man kann sogar zeigen, daß eine vollständige Enumeration durch Vermeidung von Wiederholungen besser abschneiden muß als eine Monte–Carlo–Suche.

Interpretiert man Evolution dagegen als einen kumulativen, hochgradig parallelen Siebprozeß, dessen Resultate leicht modifiziert in das nächste Sieb weitergegeben werden, erscheint die in ungefähr 4,5 Milliarden Jahren entstandene Vielfalt und Effizienz nicht mehr so wundersam. Bei der Nachahmung geht es nicht darum, diese Vielfalt durch ein Rechnermodell im Zeitraffer nachzuerfinden, sondern darum, jene Prinzipien und Mechanismen zu isolieren und auszunutzen, die ihrerseits wieder der Evolution unterlagen.

So ist fast zwangsläufig ein Mechanismus „erfunden“ worden, der es artverwandten Individuen erlaubt, Teile ihres genetischen Materials auszutauschen, um wechselnden Umweltbedingungen besser begegnen zu können. Bei Pantoffeltierchen geht dieser Vorgang übrigens nicht mit der Vermehrung einher. Selbst komplexe Umweltmodelle konnten gelernt werden: Die Durchmesserhältnisse an Verzweigungsstellen des Blutgefäßsystems liegen nahe am theoretischen, durch Naturgesetze vorgegebenen Optimum  $2^{-1/3}$ .

*Genetische Algorithmen* jenseits und *Evolutionsstrategien* diesseits des Atlantiks entstanden etwa gleichzeitig und unabhängig voneinander Mitte der 60er Jahre als globale Optimierverfahren. In beiden Verfahren „lernt“ eine Population kollektiv durch folgende Prinzipien:

- Population
- Vererbung (Rekombination bzw. Crossing-Over)
- Mutation
- Selektion

Während man Mutation und Rekombination als „Motor“ des Fahrzeuges Evolution bezeichnen kann, sorgt die Selektion für dessen „Steuerung“. Sie kann zwar nur den Phänotyp — den äußerlich sichtbaren Anpassungsgrad — bewerten, damit indirekt aber auch den das Erscheinungsbild prägenden Genotyp. Aus obiger Metapher mag auch deutlich werden, wie wichtig eine ständige Quelle neuer Varianten ist, denn ein stehendes Fahrzeug benötigt keine Steuerung! Angepaßtheit meint einen immer wieder neu zu beurteilenden Zustand, da auch die selektierende Umwelt selbst — nicht zuletzt durch die in ihr lebenden Organismen — Veränderungen unterworfen ist. Für weitere Grundlagen aus Genetik bzw. Biologie sei auf [10] verwiesen.

## 4.1 Evolutionsstrategien

Die von Rechenberg und Schwefel gemeinsam entwickelten *Evolutionstrategien* [22, 27] gleichen einem Experimentator, der ein Objekt, über das er im Extremfall nichts weiß (Black Box), anhand endlich vieler externer Regler in einen optimalen Zustand bringen

möchte. Alles, was dazu aus der Box herauskommen muß, ist ein Qualitätsmaß, so daß eine Veränderung der Einstellungen wenigstens durch „besser“ oder „schlechter“ klassifiziert werden kann. Richtung und Schrittweite der Änderungen sind für alle Optimierverfahren entscheidende Kriterien. Die folgende Abbildung macht deutlich, daß diese beiden Elemente laufend angepaßt werden müssen, wenn sich ein Erfolg einstellen soll. An der im rechten Teil der Abbildung gezeigten Zielfunktionstopologie läßt sich eine andere Metapher veranschaulichen — die des *Bergsteigers im Nebel*, der Gefahr läuft, mit Erreichen des nächstgelegenen Gipfels zufrieden zu sein, obwohl jenseits eines Tales noch höhere Gipfel liegen können.

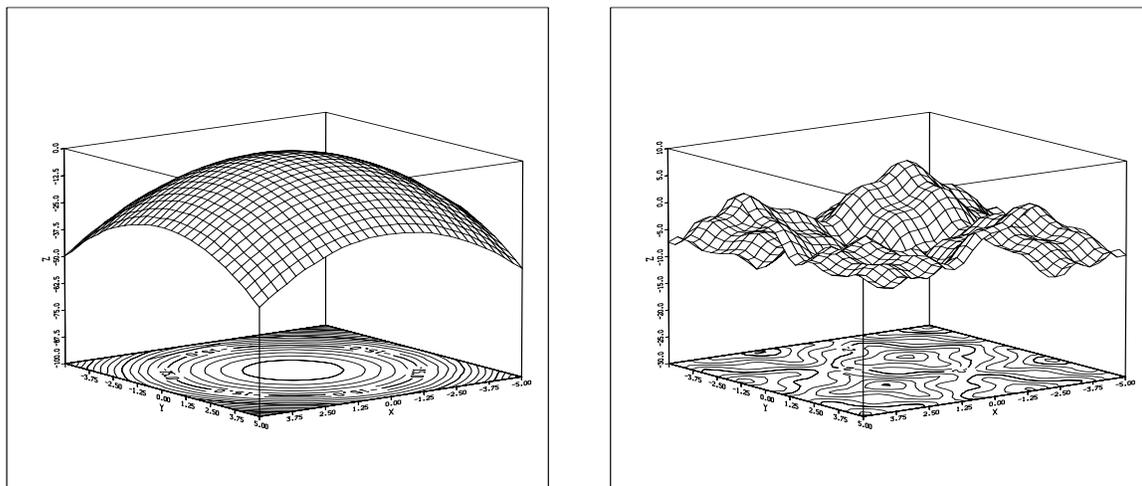


Abbildung 8: Leichte und schwierige Zielfunktionstopologie

Die ersten Versuche, Prinzipien der organischen Evolution auf einem Digitalrechner nachzuzahlen, wurden zur experimentellen Parameteroptimierung technischer Objekte wie z.B. einer Überschalldüse eingesetzt. Bei der zweigliedrigen oder  $(1 + 1)$ -Evolutionstrategie erzeugt ein Elter mit Hilfe normalverteilter Mutationen, die kleine Veränderungen wahrscheinlicher eintreffen lassen als große, solange Nachkommen, bis einer besser als der Elter ist und dessen Rolle übernimmt. Aufgrund dieser einfachen Struktur gibt es für diese Strategie theoretische Resultate über Schrittweitensteuerung und Konvergenzgeschwindigkeit: Das Verhältnis erfolgreicher zu allen Mutationen sollte  $1/5$  betragen. Diese erste einfache Variante wurde dann noch auf eine  $(\mu + 1)$ -Strategie erweitert, die erstmals aufgrund des Vorhandenseins mehrerer Eltern das Prinzip der Rekombination

umsetzen konnte. Mutation und exogene Schrittweitenkontrolle wurden unverändert in diese Variante übernommen [22].

Schwefel [27, 28] verallgemeinerte diese Verfahren auf die mehrgliedrige oder  $(\mu \mp \lambda)$ -Strategie, die jetzt genauer betrachtet werden soll. Ein *Individuum* im Rechner besteht aus folgenden „Genen“:

- Reellwertige *Objektvariablen*  $x_i$  sollen durch Rekombination und Mutation so eingestellt werden, daß ein Ziel optimal erfüllt wird.
- Reellwertige *Strategievariablen*  $\sigma_i$  steuern die Mutabilität der  $x_i$ . Sie liefern die Standardabweichung für die  $(0, \sigma)$ -Normalverteilung, die als ungerichtete Mutation die  $x_i$  additiv überlagert. Die  $\sigma_i$  rekombinieren und mutieren selbst auch von Generation zu Generation. In ihren Werten liegt das „Modell“ verborgen, das die Population sich von der Zielfunktionstopologie gemacht hat. Über Faktoren ist die Anpassungs- oder Lerngeschwindigkeit der Schrittweiten steuerbar. Die exogene Schrittweitenkontrolle ist also zugunsten einer Selbstadaption aufgegeben worden: Die Selektion wird früher oder später jene Individuen bevorzugen, die ein gutes Modell der Zielfunktion erlernt haben. Das Lernen verläuft somit auf zwei Ebenen, da sowohl Objekt- als auch Strategievariablen eingestellt werden müssen.

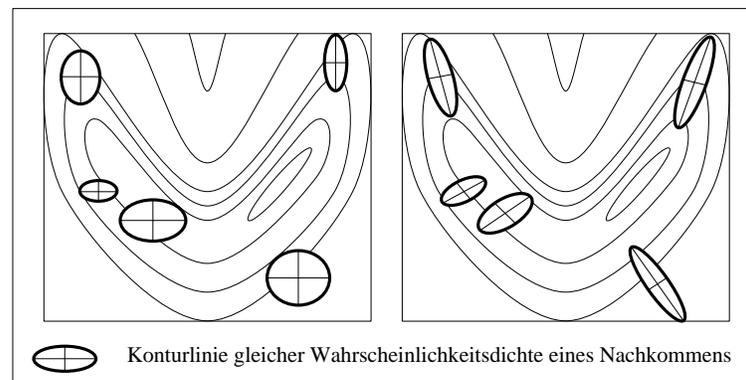


Abbildung 9: Mutationen ohne und mit Lagewinkel

Da diese Konfiguration eine Suche in Richtung der Koordinatenachsen bevorzugt, können gegebenenfalls noch vererbare und mutierbare Lagewinkel hinzugenommen

werden, um die Mutationsellipsoide frei im Raum drehen zu können (siehe Abbildung 9, entnommen aus [12]). Mathematisch verbergen sich hinter diesen Winkeln die Kovarianzen der Mutationen. In der Natur existiert ebenfalls eine zweite Ebene, z.B. in Form von Reparaturenzymen.

- Der Zielfunktionswert wird als „Phänotyp“ (Fitneß) benutzt, den die Selektion bewertet.

Von den  $\mu$  Eltern der Generation  $g$  werden  $\lambda$  Nachkommen in drei Schritten erzeugt:

- Objekt- und Strategievariablen werden gemäß gewähltem Rekombinationstyp aus zwei oder mehr Eltern zusammengesetzt, wobei alle Eltern unabhängig von ihrer Fitness gleiche Vermehrungschancen haben. Realisiert sind dabei folgende Möglichkeiten:
  - keine Rekombination,
  - diskrete Rekombination zwischen je zwei oder allen Eltern
  - intermediäre Rekombination zwischen je zwei oder allen Eltern,
  - Crossing-Over mit einer oder mehr Bruchstellen.

Der Rekombinationstyp kann für Objekt- und Strategievariablen getrennt eingestellt werden. In einer Testreihe mit 50 Funktionen hat sich herausgestellt, daß gute Resultate mit einer diskreten Rekombination für die Objekt- und einer intermediären für die Strategieparameter erzielt werden [27].

- Im zweiten Schritt mutieren zunächst die Strategievariablen, deren neue Werte dann zur Veränderung der Objektvariablen benutzt werden.
- Je nach gewählter Strategie wird dann selektiert. Bei der Plus-Strategie werden die  $\mu$  besten der  $\mu + \lambda$  Individuen in der Population zu Eltern der nächsten Generation, bei der Komma-Strategie wird nur unter den  $\lambda$  Nachkommen ausgewählt. Diese zweite Variante liegt näher am biologischen Vorbild, da kein Individuum beliebig lange überleben kann wie bei der Plus-Strategie. Gerade weil im Verlauf einer Komma-Strategie Verschlechterungen zugelassen werden, kann sich — ohne exogene Kontrolle — durch „Vergessen“ phänotypisch guter Individuen (mit eventuell in der Zukunft unangepaßten Schrittweiten) ein internes Modell der Zielfunktion in

den Strategieparametern bilden. Dies zeigt sich besonders deutlich, wenn das Optimum über der Zeit wandert (siehe Abbildung 10): Die Plus-Variante hält an einmal gefundenen guten Phänotypen fest, während die Komma-Strategie dem periodisch weiterwandernden Optimum folgt.

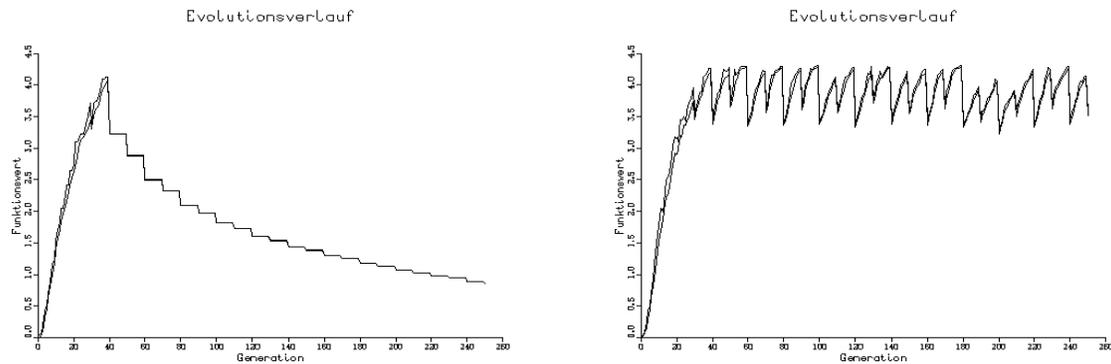


Abbildung 10: Plus- und Komma-Strategie bei wanderndem Optimum

Durch die Wahl des Verhältnisses  $\mu/\lambda$  kann man die Eigenschaft des Optimierverfahrens einstellen: Wünscht man eine schnelle lokale Konvergenz, wird man den Quotienten klein wählen, sucht man nach dem globalen Optimum, sollte man sich für eine „weichere“ Selektion entscheiden.

Evolutionsstrategien sind für die Optimierung universell geeignet, da sie nur ein Minimum an Information über die zu lösende Aufgabe benötigen — insbesondere keine Ableitungen der Zielfunktion. Sie sind in der Lage, hochdimensionale, multimodale, nichtlineare, diskrete/stetige Zielfunktionen mit linearen und/oder nichtlinearen Restriktionen zu verarbeiten. Hinter dem Zielfunktionswert kann sich auch das Resultat eines Simulationslaufes verbergen, und die Restriktionen können Festigkeitsforderungen repräsentieren, die mittels einer Finite-Element-Methode berechnet werden müssen [6]. Eine Anpassung auf Probleme mit mehrfacher Zielsetzung (Vektoroptimierung) wurde bereits durchgeführt [19], ebenso die Lösung NP-vollständiger Probleme wie das des Handlungsreisenden [2, 25]. Denkbar sind ferner experimentelle Anwendungen, bei denen ein Rechner nur Vorschläge macht und ein Mensch nach nicht formalisierbaren Kriterien die Selektion übernimmt.

Auch das Aufkommen von Parallelrechnern eröffnet neue Perspektiven, da die den Strategien innewohnende skalierbare Parallelität genutzt werden kann [13].

Eine weitere Möglichkeit, Evolutionsstrategien zu nutzen, soll zum Abschluß noch erwähnt werden: Man kann sie auch als Modelle natürlicher Prozesse interpretieren und letztere dadurch besser verstehen. „Modell“ darf allerdings hier nicht explikativ, sondern nur deskriptiv verstanden werden. Neben der ästhetischen Schönheit kann man aus den Gebilden der fraktalen Geometrie lernen, daß gerade Linien in der Natur eher die Ausnahme sind. In diesem Sinne können vielleicht auch Evolutionsstrategien durch die Demonstration von Synergieeffekten beim kollektiven Lernen einer Population einen Beitrag leisten. Das Motto müßte dann lauten: Gemeinsame Experimente statt rücksichtsloser Konkurrenz [29].

## 4.2 Genetische Algorithmen

Der Einsatzbereich der von Holland [14] entwickelten *genetischen Algorithmen* deckt sich prinzipiell mit dem der Evolutionsstrategien, und auch die grundlegenden Ideen entsprechen den in der Einleitung dieses Kapitels erwähnten natürlichen Prinzipien. Diese Verfahren greifen auf frühere Ideen von Bremermann zurück [4], der bereits eine binäre Kodierung vorschlug. Trotz der gemeinsamen Wurzeln gibt es jedoch einige Unterschiede zwischen genetischen Algorithmen und Evolutionsstrategien, die im folgenden dargestellt werden sollen [12].

- Der größte Unterschied liegt in der Kodierung der zu optimierenden Parameter durch ein (in der Regel) binäres Alphabet. Die genetische Information eines Individuums liegt als Folge von Bits (Bitstring) vor, die Black Box verfügt also nicht mehr über Regler, sondern über Ein/Aus-Schalter [9]. Zur Berechnung der Fitneß muß also häufig erst dekodiert werden. In die Dekodierung ist auch die Behandlung von Restriktionen integriert, da diese im allgemeinen zunächst einen ganzzahligen Wert ergibt, der dann in einem zweiten Schritt in ein reellwertiges, notwendigerweise beschränktes Intervall abgebildet wird. Nichtlineare Nebenbedingungen werden über Straffunktionen realisiert, die bei Verletzung einer Restriktion die Fitneß eines Individuums vermindern.

- Als genetische Operatoren sind ebenfalls Mutation und Rekombination bzw. Crossing-Over vorgesehen, allerdings mit einer anderen Rollenverteilung: Das Crossing-Over, das mit einer einstellbaren Wahrscheinlichkeit auftritt (typischerweise  $\approx 0.6$ ), spielt die Hauptrolle bei der Erzeugung von Individuen, während Mutationen, die wie in der Natur an einzelnen Genen relativ selten auftreten (Wahrscheinlichkeit  $\approx 0.001$ ) nur die Aufgabe haben, den totalen Verlust einer Null oder Eins — eines Allels — in der Population an einer bestimmten Bit-Position zu verhindern. Beide Wahrscheinlichkeiten werden exogen vorgegeben und nicht im Laufe der Generationen erlernt. Die Wahl des Gray-Code als Abbildung zwischen Bitstring und ganzer Zahl stellt sicher, daß eine Bit-Mutation nur eine Veränderung der ganzen Zahl um  $\pm 1$  bewirken kann.
- Eine intermediäre Rekombination — im einfachsten Fall die Bildung des Mittelwertes der Eltern — ist auf Bits natürlich nicht anwendbar, d.h. die Durchmischung der genetischen Information wird in Evolutionsstrategien leichter erreicht.
- Es werden keine überschüssigen Nachkommen wie bei den Evolutionsstrategien (und in der Natur) erzeugt.
- Ein anderer wesentlicher Unterschied liegt im Selektionsmechanismus. Reproduktionswahrscheinlichkeiten sind dem Anteil des jeweiligen Individuums an der Gesamtfitneß der Population proportional, können also bei Vorliegen eines gegenüber den anderen sehr guten Individuums zu einer Dominanz dieses Individuums in der Folgegeneration führen. Ein solches Verhalten kann sogar bei einfachen, unimodalen Zielfunktionen auftreten und den weiteren Fortschritt der Suche hemmen. Diese Schwäche läßt sich aber durch andere Selektionsmechanismen teilweise beheben, wie z.B. durch das Ranking, bei dem einem Individuum in Abhängigkeit von seiner relativen Fitneß (Plazierung) eine feste Anzahl von Nachkommen zugeteilt wird.

Einen Überblick über die Anwendungsgebiete genetischer Algorithmen findet man ebenso bei Goldberg [9] wie eine Einführung in *Classifier-Systeme*, in denen ein genetischer Algorithmus auf Kodierungen von Regeln arbeitet, die in ihrer Umwelt mit richtigen Vorhersagen bestehen müssen.

Die folgende Tabelle faßt die wesentlichen Unterschiede noch einmal zusammen:

<i>genetische Algorithmen</i>	<i>Evolutionsstrategien</i>
binäre Kodierung	reellwertige Variablen
eingeschränkter Suchraum wg. Dekodierung	prinzipiell unbeschränkter Suchraum
Rekombination als wichtigster Suchoperator	Mutation als wichtigster Suchoperator und Rekombination für die Schrittweitenanpassung
kein kollektives Lernen von Verfahrensparametern	kollektives Lernen von Verfahrensparametern
Lernen auf einer Ebene	Lernen auf mehreren Ebenen gleichzeitig

## 5 Zusammenfassung / Ausblick

Abschließend soll versucht werden, von einem höheren Abstraktionsniveau aus die gemeinsamen Aspekte der hier vorgestellten naturanalogen Verfahren als lediglich verschiedene Aspekte einer Grundidee zu interpretieren. So werden jeweils nur die Rahmenbedingungen (Parameter) vorgegeben, die den sich selbst überlassenen Algorithmen Lernen ermöglichen. Alle Algorithmen verfügen über eine Zufallskomponente, die bereits erreichte Zustände modifiziert und das Ergebnis der jeweiligen Selektionskomponente „vorschlägt“. Die langfristige Ordnung wird also durch eine selektive Stabilisierung „guter“ Zustände erreicht. Allen Verfahren gemeinsam ist auch das verteilte Wissen, die kollektive Intelligenz. Lernen kann daher nur durch Kooperation stattfinden. Dies unterstreichen auch Simulationen, in denen ein Rechner mit einer umfangreichen Aufgabe beauftragt wird und sich gegen einen auszuhandelnden „Preis“ von anderen Rechnern helfen lassen kann. Ohne äußere Steuerung erreichen die Rechner eine gleichmäßige Auslastung [16].

Bisher können nur die Regeln nachgebildet werden, nach denen die Materie sich organisiert oder die Evolution verläuft, ohne allerdings die Entstehung von Neuem (z.B. Wetter) ausreichend vorhersagen zu können. Läßt man sich darauf ein, aus den hier vorgestellten, relativ einfachen Modellen erheblich komplizierterer natürlicher Vorgänge Konsequenzen zu ziehen, so müßte eine lauten: Die Begriffe „Selbstorganisation“ oder „Evolution“ stellen eigentlich nur dasselbe Phänomen in einen stärker physikalischen bzw. stärker biologischen Begriffszusammenhang. Sie fassen solche Vorgänge zusammen, die in einem System mit vielen Attraktoren einen stabilen anstern und diesen auch in gewissen Grenzen gegen Störungen verteidigen können.

Ein Ziel dieses Überblicks war die Vorstellung von Verfahren, die auch noch solche Modelle optimieren können, die im Extremfall keine der üblichen mathematischen Voraussetzungen wie Stetigkeit und Differenzierbarkeit mehr erfüllen. Die größere Freiheit bei der Modellierung wird allerdings mit einem Verlust an Konvergenzsicherheit und –geschwindigkeit bezahlt. Natürlich ist das Simplex–Verfahren oder eine seiner Varianten das effizienteste, wenn eine lineare oder nichtlineare Zielfunktion sowie lineare Restriktionen vorliegen. Wird man aber den komplexen und verflochtenen Beziehungen in der Realität mit linearen Modellen wirklich gerecht? Deutet die Anpassungs– und Lernfähigkeit von Märkten nicht eher darauf hin, daß auch hier weitgehend stabile Inseln im Chaos existieren, daß also die Systeme nichtlinear und dynamisch rückgekoppelt sind? Und ist das umgebende

Chaos nicht gerade der Schlüssel zur Anpassung an veränderliche Bedingungen? Legt der letzte weltweite Börsencrash nicht den Gedanken nahe, daß hochgradig geordnete Systeme mit starren Regeln auf Störungen sehr viel empfindlicher reagieren und ins Chaotische umkippen können?

## Literatur

- [1] E. H. L. AARTS UND J. H. M. KORST, *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*, Wiley & Sons, Chichester, 1989.
- [2] P. ABLAY, *Optimieren mit Evolutionsstrategien*, Spektrum der Wissenschaft, Juli 1987, S. 104–115.
- [3] T. BARR, *Netze im Aufwind*, c't, Apr. 1991.
- [4] H. J. BREMERMAN, *Optimization through Evolution and Recombination*, in Self-Organizing Systems, M. C. Yovits, G. T. Jacobi, und D. G. Goldstein (Hrsg.), Spartan Books, Washington, D.C., 1962, S. 93–106.
- [5] J. BRIGGS UND F. D. PEAT, *Turbulent Mirror — An Illustrated Guide to Chaos Theory and the Science of Wholeness*, Harper & Row Publishers, New York, 1989.
- [6] I. CAMPOS PINTO, *Wissensbasierte Unterstützung bei der Lösung von Optimierungsaufgaben*, PhD thesis, Universität Dortmund, Fachbereich Informatik, 1989.
- [7] A. K. DEWDNEY, *Computer-Kurzweil*, Spektrum der Wissenschaft, Okt. 1988.
- [8] M. GARDNER, *The Fantastic Combinations of John Horton Conway's New Solitaire Game of 'Life'*, Scientific American, Apr. 1970.
- [9] D. E. GOLDBERG, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, Massachusetts, 1989.
- [10] L. GONICK UND M. WHEELS, *Genetik in Cartoons*, Paul Parey, Berlin und Hamburg, 1986.
- [11] D. O. HEBB, *The Organization of Behaviour*, Wiley & Sons, New York, 1949.
- [12] F. HOFFMEISTER UND T. BÄCK, *Genetic Algorithms and Evolution Strategies: Similarities and Differences*, in Parallel Problem Solving from Nature, Proceedings of the 1<sup>st</sup> PPSN-Workshop, Dortmund, 1990, H.-P. Schwefel und R. Männer (Hrsg.), Jahrgang 496 *Lecture Notes in Computer Science*, Berlin, 1991, Springer, S. 445–469.
- [13] F. HOFFMEISTER UND H.-P. SCHWEFEL, *A Taxonomy of Parallel Evolutionary Algorithms*, in PARCELLA '90, G. Wolf, T. Legendi, und U. Schendel (Hrsg.), Berlin, 1990, Akademie-Verlag, S. 97–107.
- [14] J. H. HOLLAND, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Michigan, 1975.
- [15] J. J. HOPFIELD, *Neural Networks and Physical Systems with Emergent Collective Computational Abilities*, in Proc. Nat. Acad. Sci. USA, 1982, S. 2554–2558.

- [16] B. A. HUBERMAN, *The Ecology of Computation*, Studies in Computer Science and Artificial Intelligence, North-Holland, Amsterdam, 1988.
- [17] S. KIRKPATRICK, C. D. GELATT, UND M. P. VECCHI, *Optimization by Simulated Annealing*, Science, Jahrgang 220, 1983, S. 671–680.
- [18] T. KOHONEN, *Self-Organization and Associative Memory*, Springer, Berlin, 1984.
- [19] F. KURSAWE, *A variant of evolution strategies for vector optimization*, in Parallel Problem Solving from Nature, Proceedings of the 1<sup>st</sup> PPSN-Workshop, Dortmund, 1990, H.-P. Schwefel und R. Männer (Hrsg.), Jahrgang 496 *Lecture Notes in Computer Science*, Berlin, 1991, Springer, S. 193–197.
- [20] M. MINSKY UND S. PAPERT, *Perceptrons*, MIT Press, Cambridge, Massachusetts, 1969.
- [21] Y. H. PAO, *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley, Reading, Massachusetts, 1989.
- [22] I. RECHENBERG, *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Frommann-Holzboog, Stuttgart, 1973.
- [23] H. RITTER, T. MARTINETZ, UND K. SCHULTEN, *Neuronale Netze*, Addison-Wesley, Bonn, 1990.
- [24] F. ROSENBLATT, *Principle of Neurodynamics*, Spartan, New York, 1962.
- [25] G. RUDOLPH, *Global Optimization by means of Distributed Evolution Strategies*, in Parallel Problem Solving from Nature, Proceedings of the 1<sup>st</sup> PPSN-Workshop, Dortmund, 1990, H.-P. Schwefel und R. Männer (Hrsg.), Jahrgang 496 *Lecture Notes in Computer Science*, Berlin, 1991, Springer, S. 209–213.
- [26] D. E. RUMELHART, G. E. HINTON, UND R. J. WILLIAMS, *Learning Internal Representations by Error Propagation*, in Parallel Distributed Processing I: Foundations, D. E. Rumelhart und J. L. McClelland (Hrsg.), MIT Press, Cambridge, Massachusetts, 1986, ch. 8.
- [27] H.-P. SCHWEFEL, *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*, Birkhäuser, Basel, 1977.
- [28] ———, *Numerical Optimization of Computer Models*, Wiley & Sons, Chichester, 1981.
- [29] ———, *Collective Phenomena in Evolutionary Systems*, in 31<sup>st</sup> Annual Meeting of the International Society for General System Research, Budapest, 1987, S. 1025–1033.
- [30] P. J. M. VAN LAARHOVEN UND E. H. L. AARTS, *Simulated Annealing: Theory and Applications*, D. Reidel, Dordrecht, 1987.
- [31] J. VON NEUMANN, *Theory of Self-Reproducing Automata*, University of Illinois Press, 1966.