

Künstliche Evolution als Modell für natürliche Intelligenz

Frank Kursawe
Hans-Paul Schwefel

Universität Dortmund
Fachbereich Informatik
Lehrstuhl für Systemanalyse
44221 Dortmund

INHALTSVERZEICHNIS

1	Einleitung	2
2	Spezialisierte Optimierverfahren und ihre Schwierigkeiten	3
3	Zu einfache Evolutionsmodelle	6
4	Die zweigliedrige Evolutionsstrategie	9
5	Die mehrgliedrige Evolutionsstrategie ...	10
5.1	...als universelles Optimierverfahren	13
5.2	...als selbstorganisierendes Optimierverfahren	14
6	Abgrenzung zu anderen naturanalogen, selbstorganisierenden Verfahren	18
7	Ausblick	20

In dem Bemühen, sich technische Hilfsmittel für vielerlei Zwecke zu schaffen, hat der Mensch wohl seit seinem Erscheinen auf dem Planeten Erde — oftmals unbewußt — natürliche Vorbilder benutzt. Es bedurfte aber erst des besseren Verständnisses für den Entwicklungsprozeß, dem Lebewesen im Laufe ihrer Geschichte unterliegen, um zu mutmaßen, daß Formen, Strukturen und Prozesse in der Natur besonders gut an ihre Umwelt angepaßt, ja eventuell sogar *optimal* seien. In einigen Fällen (siehe z.B. [22]) konnte dies inzwischen sogar nachgewiesen werden. Bionik stellt heute konsequenterweise den Versuch dar, dieses Reservoir hervorragender Lösungen für Aufgaben, wie sie in der Technik auch auftreten, bewußt zu nutzen.

Der unvorstellbar lange Zeitraum von 3,5 Milliarden Jahren und die riesige Oberfläche der Erde im Vergleich zur Größe und Lebensdauer einzelner Lebewesen leisteten lange Zeit — und zum Teil heute noch — der intuitiven Vorstellung Vorschub, die natürliche Strategie der Lösungsfindung sei einer Monte-Carlo-Strategie des enumerativen Durchspielens aller Möglichkeiten ähnlich. Dabei bedarf es nur einer groben Überschlagsrechnung, um schnell festzustellen, daß dies unmöglich so funktioniert haben kann: Die Keimzelle eines Menschen beherbergt in ihrem Zellkern etwa 50.000 Gene. Jedes Gen besteht im Mittel aus etwa 300 Triplets von Nukleotidbasen, derer es vier verschiedene Arten gibt. Selbst unter Berücksichtigung der Redundanz des genetischen Codes, welcher mittels $4^3 = 64$ Tripletsorten nur 20 verschiedene Aminosäuren codiert, verbleiben noch $20^{15.000.000} \approx 10^{19.500.000}$ Kombinationen für den Genotyp. Legt man die heutige Populationszahl von etwa 5 Milliarden Menschen zugrunde, so bedürfte es immer noch $10^{19.499.990}$ Generationen, um alle Kombinationen auch nur einmal ‚durchzuspielen‘. Argumente dieser Art mußten wiederholt herhalten, um die Unmöglichkeit von Evolution zu ‚beweisen‘.

Die andere Möglichkeit, alternative Modelle des evolutionsstrategischen Prozesses zugrunde zu legen, nahmen u.a. Bremermann [5], Holland [15] sowie Rechenberg [21] und Schwefel [26] unter die Lupe, indem sie diese herkömmlichen Optimierverfahren gegenüberstellten. Die ersten, in den sechziger Jahren unternommenen Versuche stießen auf wenig Gegenliebe, wobei hier nicht versucht werden soll, über die Gründe zu spekulieren. Erst heute, mit dem Erscheinen von echten Parallelrechnern — und vielleicht auch mit der wachsenden Ernüchterung bei dem Versuch, menschliche Intelligenz durch maschinelles Schließen nachzuahmen bzw. ein umfassendes Modell der Welt im Rechner nachzubilden und zu manipulieren, um intelligente Rechenprozesse zu kreieren — rücken *Genetische Algorithmen* (à la Holland) und *Evolutionsstrategien* (à la Rechenberg und Schwefel) erneut in das Zentrum des Interesses. Dies belegen die seit 1985 alle zwei Jahre in den USA stattfindenden ICGA (International Conferences on Genetic Algorithms), die seit 1990 alle zwei Jahre in Europa bzw. diesseits des Atlantiks ausgerichteten PPSN-Konferenzen (Parallel Problem Solving from Nature) und auch die Reihe der „Artificial Life“-Konferenzen.

Während Genetische Algorithmen bei der Modellierung der Evolution auf der Genotyp-Ebene ansetzen, ja sogar mit binärer Kodierung der Variablen arbeiten, operieren Evolu-

tionsstrategien eher auf der Phänotyp-Ebene, d.h. mit reellwertigen Variablen. Während die älteste Version der Evolutionsstrategien (zweigliedrige Strategie) nur Mutation und Selektion nachahmt und erst in der erweiterten Form (mehrgliedrige Strategie) auch das Populationsprinzip und die Rekombination genutzt wird, basierten die Genetischen Algorithmen (siehe Kapitel 6) von Anfang an auf allen vier bisher genannten Evolutionsmechanismen bzw. -prinzipien (eine neuere Zusammenfassung stammt von Goldberg [11]).

Im folgenden soll insbesondere der bisher nur auf der Basis der Evolutionsstrategien verfolgte Aspekt der *populativen Selbstanpassung* von strategischen Größen — wie der passenden Mutationsraten — hervorgehoben werden. Mit exogen festgelegten Mutationsraten oder -schrittweiten, Rekombinationsparametern oder Selektionsregeln entfalten evolutionäre Algorithmen (Sammelbegriff für Genetische Algorithmen und Evolutionsstrategien) nur begrenzte Wirksamkeit. Sie müssen vom Benutzer auf die im speziellen Fall günstigen Werte voreingestellt werden. In der Natur gibt es aber keine Instanz, welche solch ein ‚Tuning‘ vornehmen könnte. Vielmehr ist anzunehmen, daß die ‚Strategie der Evolution‘, so wie sie heute zu beobachten ist, ebenfalls evolvierte. Zum Beispiel wurde die geschlechtliche Vererbung erst zu einem späteren Zeitpunkt ‚erfunden‘. Ihre allgemeine Durchsetzung — trotz höherer Kosten — muß durch den Erfolg begründet sein, welcher durch die Rekombination erzielt wurde.

Zunächst soll aber erst einmal aufgezeigt werden, warum es einen Bedarf an derartigen heuristischen Optimum-Suchverfahren überhaupt gibt.

2 SPEZIALISIERTE OPTIMIERVERFAHREN UND IHRE SCHWIERIGKEITEN

Unter dem Begriff *Optimierverfahren* faßt man üblicherweise jene Algorithmen zusammen, die in der Lage sind, einen besonders vorteilhaften Zustand — das *Optimum* — des gerade betrachteten Systems aufzufinden. Dieses System muß allerdings einige Voraussetzungen erfüllen, damit Verfahren aus dem Bereich der *Parameteroptimierung*, auf die wir uns hier beschränken wollen, eingesetzt werden können:

- Das System muß einstellbare Größen (Inputs, exogene Parameter) besitzen.
- Es muß ein Auswahlkriterium zwischen verschiedenen, bewertbaren Lösungen geben — im Extremfall nur die qualitative Aussage ‚besser‘ bzw. ‚schlechter‘.

Mathematisch formuliert, stellt sich ein nicht-diskretes Parameter-Optimierungsproblem wie folgt dar, wobei man sich wegen ($\max f = -\min(-f)$) auf Minimierungsprobleme beschränken kann:

$$\begin{array}{ll}
 \text{minimiere} & f : \mathbb{R}^n \rightarrow \mathbb{R}, \\
 \text{ggf. mit den Restriktionen} & g_j \geq 0, \quad j = 1, \dots, m; \\
 \text{gesucht:} & \underline{x}^* \in \mathbb{R}^n, \\
 \text{so daß gilt:} & \forall \underline{x} \in \mathbb{R}^n : f(\underline{x}^*) \leq f(\underline{x}) \\
 \text{und ggf.} & g_j(\underline{x}^*) \geq 0, \quad j = 1, \dots, m
 \end{array}$$

Die Aufgabe, ein gegebenes System in den global optimalen Zustand zu bringen, ist im allgemeinen praktisch *unlösbar*. Die nächstliegende Methode, den Suchraum einfach zu durchmustern (*Rastersuche*) und sich den jeweils besten aufgetretenen Wert zu merken, scheitert schon bei relativ kleinen Problemen am exponentiell wachsenden Suchaufwand, oder wie Bellman es formuliert hat, am „Fluch der Dimension“.

Daher versucht man, das allgemeine Problem auf Spezialfälle einzuschränken, für diese — falls möglich — eine Theorie zu entwickeln und daraus ein Optimierverfahren abzuleiten. So ist eine Fülle von mehr oder weniger spezialisierten Optimierverfahren entstanden, die gut funktionieren, falls das System alle Voraussetzungen erfüllt, die aber versagen können oder sich zumindest als ineffizient erweisen, falls erwartetes ‚internes Modell‘ und System, oft auf die Form einer Zielfunktion $\underline{x} \rightarrow f(\underline{x})$ reduziert, nicht übereinstimmen.

Optimierverfahren müssen sich im allgemeinen aufgrund fehlender Informationen über die inneren Zusammenhänge des zu optimierenden Systems (‚Black Box‘) mit Heuristiken behelfen, d.h. sie nutzen extern vorgegebene Informationen über erfolgreiche Suchrichtungen, oder sie orientieren sich am Erfolg und Mißerfolg bereits durchgeführter Suchschritte, um zum Optimum vorzustoßen.

Der bekannteste Vertreter mehrdimensionaler Optimierverfahren ist wohl die *Simplex-Methode* (inklusive ihrer Modifikationen), die für einen durch lineare Restriktionen eingeschränkten Suchraum und eine lineare (bzw. höchstens quadratische) Zielfunktion relativ schnell und sicher arbeitet [8]. Mathematisch gesprochen, optimiert man eine höchstens quadratische (und damit konvexe) Zielfunktion über einem ebenfalls konvexen Suchraum. Die strengen Voraussetzungen stellen sicher, daß ein gefundenes lokales immer auch das globale Optimum ist; Zielfunktionen in dieser extrem eingeschränkten Welt verhalten sich ‚gutmütig‘.

Vertreter aus der Klasse der *Gradientenverfahren* sind an diese lineare Welt nicht mehr gebunden, doch sie ‚glätten‘ die raue Welt durch eine andere mathematische Voraussetzung: Benötigt werden vor dem Start des Verfahrens explizit die ersten partiellen Ableitungen der Zielfunktion, also Informationen über lokale Steigungen. Diese Strategien legen also ein internes, lokal lineares Modell des Systems zugrunde. Die Suchrichtung orientiert sich am steilsten An- bzw. Abstieg der Funktion.

Das *Newton-Verfahren* benötigt sogar zusätzlich die zweiten partiellen Ableitungen (Hesse-Matrix), also Informationen über lokale Krümmungen, verfügt aber damit über ein internes quadratisches Modell seiner Umwelt. Die *Quasi-Newton-Verfahren* beschaffen sich die nötigen Informationen approximativ im Laufe der Suche selbst, ebenso die *konjugierten Gradienten-* und *variable Metrik-Verfahren*.

In die Welt der bisher vorgestellten Verfahren passen keine Verschlechterungen, so daß die Suche nach dem Optimum abgebrochen werden kann, falls antizipierte Verbesserungen ausbleiben. Bei einer Zielfunktionstopologie wie in Abbildung 1 (links) führt dies auch zum gewünschten Ergebnis, bei der rechts dargestellten Topologie wird man aber nur das dem Startpunkt nächstgelegene lokale Optimum finden und dort ‚steckenbleiben‘. Das Newton-

Raphson–Verfahren kann sogar divergieren, wenn eine Diskrepanz zwischen internem Modell und der Realität auftritt.

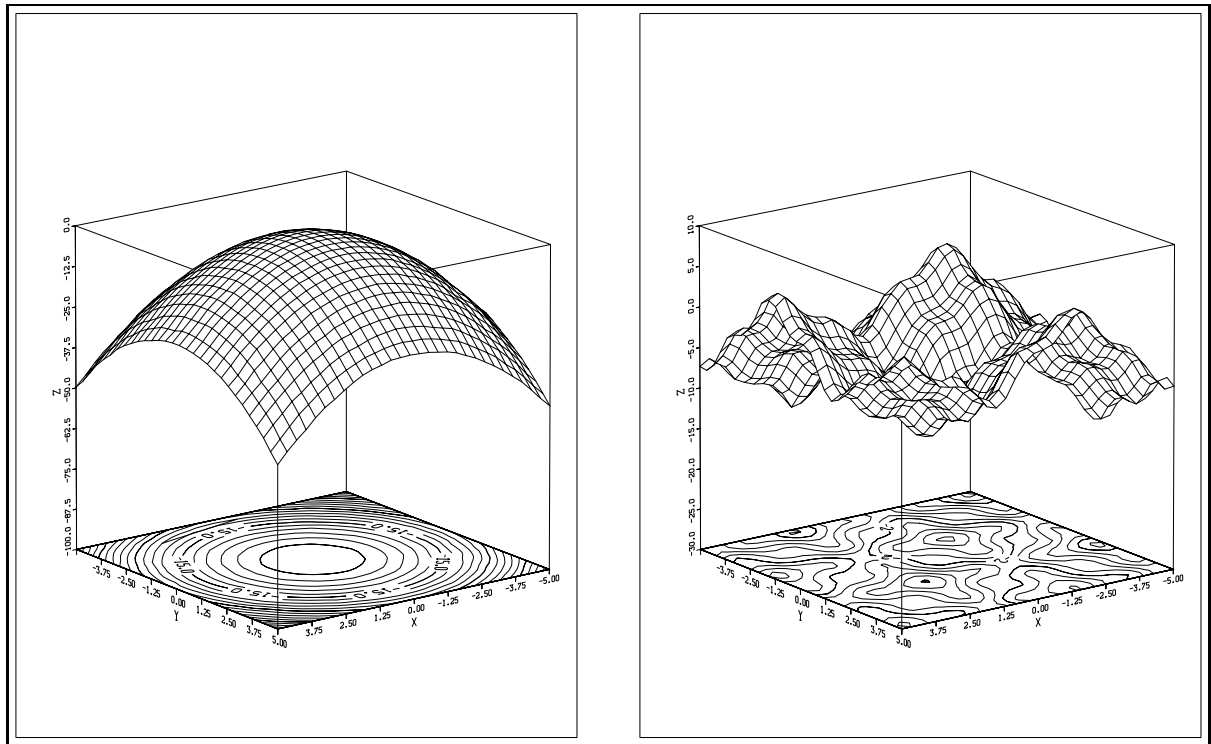


Abbildung 1: Leichte und schwierige Zielfunktionstopologie

Die bisher vorgestellten *deterministischen* Strategien sind natürlich anderen Verfahren überlegen, wenn die Aufgabenstellung in das Schema ihrer Modellannahmen paßt. So sind Newton–Raphson–Verfahren in der Lage, das Optimum einer quadratischen Zielfunktion innerhalb einer Iteration zu finden, wenn auch dieser eine Schritt durch eine Matrixinversion sehr aufwendig wird. Für den anderen Fall sind einige unkonventionelle Strategien entworfen worden, darunter auch *stochastische* (siehe Kapitel 3, 4, 5 und 6).

Polyeder–Verfahren (*Simplex–* oder *Complex–Strategie*) für die nichtlineare Optimierung haben sich in einem umfangreichen Vergleichstest als robust in schwierigen Situationen erwiesen, ebenso wie das *Pattern Search–Verfahren* oder die Methode der *rotierenden Koordinaten* [26, 27]. Interessanterweise suchen viele Strategien in ‚pathologischen‘ Situationen Zuflucht beim Zufall, d.h. wenn der Vorrat an deterministischen Regeln kein Weiterkommen mehr ermöglicht.

Bei technischen Optimierproblemen wird man relativ selten das System mittels einer geschlossenen, analytischen Zielfunktion der Form $\underline{x} \rightarrow f(\underline{x})$ beschreiben können. Oft benötigt

man ein *Simulationsmodell*, um eine betriebliche Wirklichkeit hinreichend genau zu erfassen. Und wenn ein solches Modell einmal erstellt und validiert ist, liegt der Wunsch nahe, über die Simulation des laufenden Betriebes hinaus nach ‚besseren‘ Zuständen des Modells und damit auch der realen Produktion zu suchen, ohne allerdings in den laufenden Betrieb einzugreifen.

Dies kann aber mit Hilfe der bisher beschriebenen konventionellen Optimierverfahren nur geschehen, wenn man sich bereits in der Modellierungsphase an den Erfordernissen eines oder mehrerer der bisher vorgestellten Verfahren orientiert hat. Bei einer komplexen Realität wird man in einen Zwiespalt zwischen der Urbildtreue des Modells und den notwendigen Voraussetzungen für eine spätere Optimierung geraten. Die Kategorien linear/quadratisch/nichtlinear für die Zielfunktion sowie ggf. linear/nichtlinear für die Restriktionen werden oft einer komplexeren Realität nicht gerecht, so z.B. bei Klima- oder Verbrennungsprozessen sowie Roboter- oder Fahrzeugsteuerungen. Die entsprechenden Modelle werden in der Regel nicht einmal die elementare Voraussetzung der Stetigkeit (kleine Änderungen an den Eingängen bewirken auch nur kleine Änderungen der Ausgänge) erfüllen, von einer stetigen Differenzierbarkeit ganz zu schweigen. In folgenden Fällen muß man mit dem Versagen der klassischen Optimierverfahren rechnen:

- bei stochastisch gestörten Modellgrößen, die in der experimentellen Optimierung den Normalfall darstellen,
- bei nichtlinear-dynamischen Modellen, die ‚deterministisch-chaotisches‘ Verhalten, also sich *scheinbar* stochastisch verändernde Modellgrößen, hervorbringen können,
- bei ‚schwierigen‘ Computermodellen, die nicht stetig und damit auch nicht differenzierbar sind,
- bei über der Zeit wandernden Optima, wie sie in Prozeßsteuerungs- und -kontrollaufgaben auftreten,
- bei Problemen aus dem Bereich der diskreten Optimierung, deren Probleme im allgemeinen *NP-vollständig* sind, d.h. die Rechenzeit für das Auffinden einer garantiert global optimalen Lösung wächst exponentiell mit der Problemgröße. So ist z.B. das Handlungsreisenden-Problem bereits für 30 Städte mit der heutigen Rechnertechnologie nicht mehr exakt lösbar. Maschinenbelegungs- oder Containerpackungsaufgaben gehören zu den praktisch relevanten diskreten Entscheidungsproblemen.

Damit sich ein Ingenieur jedoch nicht schon in der Modellierungsphase der normativen Kraft des leicht Optimierbaren unterwerfen und somit auf Realitätsnähe verzichten muß, wurden Optimierverfahren entwickelt (siehe Kapitel 4, 5 und 6), die universell einsetzbar, also ‚anspruchlos‘ sind, diese Eigenschaft aber mit einem Verzicht auf die Garantie des Auffindens eines globalen Optimums und einer langsameren Konvergenzgeschwindigkeit — verglichen mit jeweils für den Spezialfall entwickelten Verfahren — bezahlen.

3 ZU EINFACHE EVOLUTIONSMODELLE

Ein uraltes, leider immer noch nicht ganz ausgerottetes, teilweise absichtliches *Mißverständnis der Evolution* besteht darin, Mutationen — den Motor des Geschehens (neben der ent-

wicklungsgeschichtlich jüngeren Rekombination) — als ‚reinen‘ Zufallsprozeß auf der phänotypischen Ebene zu deuten und zu modellieren. Innerhalb der möglichen Bandbreite der Ausprägung eines jeden Merkmals hat demnach ein Nachkomme eine Position, die völlig unabhängig von der seiner Eltern ist — mit gleicher Wahrscheinlichkeit für jede Position. Die Wahrscheinlichkeit, die optimale Position mit der Genauigkeit d bei einer Bandbreite von D zu ‚erwischen‘, wäre hierbei

$$p_1 = \frac{d}{D},$$

die Wahrscheinlichkeit, danebenzuliegen

$$\bar{p}_1 = 1 - \frac{d}{D},$$

unabhängig von der Vorgeschichte. Ein allmähliches Annähern an den besten Zustand im Sinne eines evolutionären Lernens ist dabei völlig ausgeschlossen. Wenn nun nicht nur ein Merkmal (Parameter) beteiligt ist, sondern n verschiedene Merkmale, dann gilt entsprechend für die Trefferwahrscheinlichkeiten

$$p_n = \left(\frac{d}{D}\right)^n$$

$$\bar{p}_n = 1 - \left(\frac{d}{D}\right)^n.$$

Wiederholt man solch ein reines Zufallsexperiment N -mal — man nennt dies eine Monte-Carlo-Strategie — so ist die Wahrscheinlichkeit, keinen einzigen Zufallstreffer zu erzielen

$$\bar{w}_N = \bar{p}_n^N = \left[1 - \left(\frac{d}{D}\right)^n\right]^N,$$

und dementsprechend die Wahrscheinlichkeit, mindestens einen Treffer zu landen

$$w_N = 1 - \bar{w}_N = 1 - \left[1 - \left(\frac{d}{D}\right)^n\right]^N.$$

Daraus ergibt sich bei vorgegebener Trefferwahrscheinlichkeit w_N die notwendige Anzahl von Experimenten (Iterationen) zu

$$N = \frac{\log(1 - w_N)}{\log\left[1 - \left(\frac{d}{D}\right)^n\right]},$$

die für eine hinreichend hohe Parameter-Anzahl $n \gg 1$ mit

$$N \approx -\log(1 - w_N) \left(\frac{D}{d}\right)^n$$

approximiert werden kann.

Schon bei einer Erfolgswahrscheinlichkeit von nur 63% und einer relativen Treffergenauigkeit von nur $\varepsilon = d/D = 10\%$ ergibt sich als erforderliche Versuchsanzahl

$$N_{63} \approx 1 \cdot 10^n, \text{ für } w_N = 90\% \text{ sogar } N_{90} \approx 2,3 \cdot 10^n,$$

in jedem Fall also ein exponentiell mit der Zahl der Variablen ansteigender Aufwand.

Wie man leicht zeigen kann, liefert hier eine systematische Suche (äquidistante Rastermethode) nach dem Optimum das gleiche Ergebnis (mit Wahrscheinlichkeit Eins) nach

$$N_{100} = 1 \cdot 10^n$$

Stichproben, die gleichmäßig über den Suchraum verteilt werden.

Diese Analyse liefert willkommene Munition

- erstens gegen den ‚Glauben‘, diese ‚Evolutionstheorie‘ könnte innerhalb von 3,5 Milliarden Jahren Entwicklungszeit die Existenz des Menschen erklären oder wenigstens als halbwegs wahrscheinlich erscheinen lassen,
- und zweitens gegen ‚Zufall als Methode‘ schlechthin.

Das erste Argument kann man nur so wenden, daß hier die Realität (nur 10^{17} Sekunden standen zur Verfügung für die ‚richtige‘ Einstellung von etwa 50.000 Genen, gebildet aus durchschnittlich je 1.000 Nukleotidbasen mit 4 Möglichkeiten) nicht mit der ‚Theorie‘ in Einklang gebracht werden kann. Statt daraufhin Evolution als Faktum abzulehnen, kann man aber auch auf die Suche nach einem besseren Evolutionsmodell gehen.

Das zweite Argument auszuhebeln, ist für viele leichter, die ‚rational-deterministische‘ Suchstrategien lieber mögen, aber scheinbar weniger offensichtlich. Eine zufällige Reihenfolge der Experimente beim Raster-Suchverfahren verschlechtert das Ergebnis keineswegs. Die ursprüngliche Monte-Carlo-Methode ist nur deswegen weniger effizient, weil sie nicht verhindert, daß unnötige Wiederholungen bzw. im Sinne der geforderten Treffergenauigkeit unnötig dicht benachbarte Experimente ausgeführt werden. Es wäre ein leichtes, die Monte-Carlo-Methode so abzuändern, daß in der ε -Nachbarschaft schon durchgeführter Experimente keine weiteren Versuche plaziert werden.

Das verführerische Gegenargument der verschwenderischen Natur, die ja so viele Experimente zeitlich parallel ausführen kann und so viel Zeit hatte, verblaßt schnell, wenn man in das obige Kalkül reale Zahlenwerte einsetzt und mit den Möglichkeiten dieser Welt vergleicht: Eine zwei Meter dicke Schicht von Bakterien (oder Hefepilzen) rund um die ganze Erde enthielte nicht mehr als etwa 10^{34} Individuen. Bei einer Generationsdauer von etwa einer Stunde hätten somit im Verlaufe von 3,5 Milliarden Jahren nicht mehr als $3 \cdot 10^{47}$ Experimente durchgeführt werden können, was — abgesehen von den Kommunikationsschwierigkeiten für den gesamten Versuch — nur dazu ausgereicht hätte, 79 Nukleotidbasen ‚richtig‘ aneinanderzureihen. Selbst ein Bakterium verfügt aber über 3.500 Gene bzw. etwa einen Million Nukleotidpaare im Chromosom.

Die Erde erweist sich also als viel zu klein und die Zeit seit dem Erscheinen erster Lebewesen als viel zu kurz, um eine Evolution à la Monte-Carlo zu ermöglichen. Vor allem die Zeitknappheit (nur 10^{17} Sekunden) muß einen enormen Druck zum ‚Erfinden‘ eines geeigneten Evolutionsmechanismus hervorgerufen haben.

So absurd, wie hier dargestellt, ist es wohl aber auf den ersten Blick nicht, die Monte-Carlo-Strategie als Modell der biologischen Evolution in Anspruch zu nehmen. Die von Ashby am Beginn des Computer-Zeitalters in seinem Buch „Design for a Brain“ [2] vorgeschlagene Strategie für einen Homöostaten, der mittels Monte-Carlo-Proben seinen Gleichgewichtszustand wiederfinden sollte, arbeitet nach eben jenem Prinzip. Aber selbst mit Supercomputern, welche bald eine Billion Floating-Point-Operationen pro Sekunde ausführen, erweist sich die kombinatorische Enumeration schon für relativ kleine Probleme als nicht mehr durchführbar.

Eine bessere, aber immer noch nicht hinreichend gute — inzwischen fast vergessene — Idee für eine Evolutionsstrategie stammt ebenfalls aus den sechziger Jahren, von Bremermann [5]. Ausgehend von einem Startpunkt im \mathbb{R}^n sollen (gleichzeitig) in mehreren Zufallsrichtungen herkömmliche eindimensionale Suchen das jeweilige relative Optimum in diesen Richtungen bestimmen. Durch Rekombination der besonders guten Endpunkte werden neue Startpunkte für weitere eindimensionale Suchen gewonnen. Bremermann untersuchte die Eignung des Verfahrens für Aufgaben der Linearen Programmierung, bei denen wegen der Konvexität des Lösungsraumes die Rekombination als eine Art Mittelwert-Bildung stets zulässige neue Startpunkte liefert. Das Verfahren erwies sich aber als der Simplex-Methode von Dantzig unterlegen und starb daher schnell aus. Immerhin zeigte es die Bedeutung des dritten Prinzips der Evolution, der Rekombination, auf, wohingegen die Monte-Carlo-Methode ja neben Mutation nur noch das Prinzip Selektion kennt, das abschließende Auswählen des besten Ergebnisses aus allen Versuchen. Bremermanns Konzept berücksichtigte wenigstens das sequentielle Generationsschema der Evolution mit Zwischenspeicherung der bisher besten (überlebendigen) Resultate und das Faktum, daß Mutationen nicht ganz unabhängig von der Vorgeschichte des Suchprozesses sind — ein wesentliches Merkmal aller ‚echten‘ Lernstrategien.

4 DIE ZWEIGLIEDRIGE EVOLUTIONSSTRATEGIE

Die erste erfolgreiche Umsetzung eines Mutations-Selektions-Schemas auf einem Digitalrechner war die *zweigliedrige* oder auch $(1 + 1)$ -*Evolutionstrategie* [21]. Ihr Erfolg lag vor allem darin begründet, daß das Würfelschema für die Mutationen dem kumulativen, hochgradig parallelen Siebprozeß in der Natur, der Erreichtes leicht modifiziert in das nächste ‚Sieb‘ schickt, schon etwas näher kam. Erst so kann man von einem schrittweisen Lernprozeß im Sinne einer Anpassung oder Meliorisierung sprechen. Eingesetzt wurde das neue Verfahren zunächst zur experimentellen Parameteroptimierung technischer Objekte wie z.B. einer Überschalldüse.

Das Ablaufschema des Algorithmus' läßt sich vereinfacht wie folgt darstellen: Ein Elter erzeugt mit Hilfe normalverteilter Mutationen, die kleine Veränderungen in den Entscheidungsvariablen wahrscheinlicher eintreffen lassen als große, solange Nachkommen, bis einer besser als der Elter ist und dessen Rolle übernimmt. Für zwei grundverschiedene Zielfunktionen konnten aufgrund dieser einfachen Struktur Aussagen über Schrittweitensteuerung und damit Konvergenzgeschwindigkeit bewiesen werden: Das optimale Verhältnis erfolgreicher zu allen Mutationen beträgt unabhängig von der Variablenzahl etwa $1/5$. Liegt die beob-

achtete Erfolgsrate darunter, sind die Schritte zu groß und sollten verkleinert werden; liegt die Erfolgsrate aber höher, können größere Schritte gemacht werden. Diese *externe Schrittweitensteuerung* wurde auch in der Erweiterung auf eine $(\mu + 1)$ -Strategie beibehalten, die aufgrund des Vorhandenseins mehrerer Eltern erstmals auch eine Rekombination des genetischen Materials erlaubte [21]. Die Konvergenzgeschwindigkeit einer $(1 + 1)$ -Strategie ist im linearen Fall etwa proportional $1/\sqrt{n}$, ein experimentelles Gradientenverfahren liegt etwa bei $1/n$, so daß für $n \geq 3$ die Evolutionsstrategie bereits schneller konvergiert.

Obwohl dieses neue Optimierverfahren keinerlei Voraussetzungen bezüglich mathematischer Eigenschaften der Zielfunktion mehr machte, also *kein* festes internes Modell mehr besaß und einen neuen, der Biologie abgeschauten Ansatz darstellte, war es dennoch ein Kind seiner Zeit und wurde auch von anderen in ähnlicher Form ohne Bezug auf das biologische Vorbild vorgeschlagen:

- Die für Optimierverfahren übliche Iterationsvorschrift findet sich in der zweigliedrigen Evolutionsstrategie wieder:

$$\underline{x}^{(g+1)} := \underline{x}^{(g)} + s^{(g)} \cdot \underline{v}^{(g)},$$

wobei g den Iterationsindex, \underline{x} den Vektor der Objektvariablen, s die Schrittweite und \underline{v} den auf die Länge 1 normierten Richtungsvektor repräsentiert.

- Die Steuerung der einen, alle Variablen verändernden und die Konvergenzgeschwindigkeit bestimmenden Schrittweite wurde exogen vorgenommen, obwohl es in der natürlichen Evolution keinen externen Beobachter gibt.
- Obwohl im Vorbild Natur einem Individuum nur eine endliche Lebenszeit vergönnt ist, verbietet die zweigliedrige Evolutionsstrategie das ‚Vergessen‘ bereits erreichter, guter Lösungen.
- Das Prinzip Selektion wird als „Survival of the Fittest“ (Singular!) interpretiert.

Trotz der neuen Idee gehört die zweigliedrige Evolutionsstrategie zu den stochastischen Gradientenverfahren, besitzt also auch eher lokale Konvergenzeigenschaften.

5 DIE MEHRGLIEDRIGE EVOLUTIONSSTRATEGIE . . .

Das in Kapitel 4 vorgestellte Verfahren wurde dahingehend erweitert, daß sich eine echte Population, bestehend aus μ Eltern und λ Nachkommen in jeder Generation, auf die Suche nach dem Optimum begibt [26, 27]. Bevor in den beiden folgenden Unterkapiteln spezielle Aspekte der *mehrgliedrigen* oder $(\mu \ddagger \lambda)$ -*Evolutionsstrategie* beleuchtet werden, soll zunächst der Algorithmus selbst erläutert werden.

Ein ‚Individuum‘ im Rechner verfügt über die folgende genetische Ausstattung:

- Reellwertige *Objektvariablen* x_i ($i = 1, \dots, n$), die von dem Verfahren so einzustellen sind, daß eine gegebene n -dimensionale Aufgabe (möglichst) optimal gelöst wird.
- Reellwertige *Strategievariablen* σ_i ($i = 1, \dots, n$) steuern die Mutabilität der x_i : Jedes der x_i wird mit einer (ungerichteten) $(0, \sigma_i)$ -Normalverteilung überlagert. Diese

Standardabweichungen σ_i rekombinieren und mutieren selbst auch von Generation zu Generation. Ihre Werte stellen das ‚Modell‘ der Zielfunktionstopologie oder ‚Umwelt‘ dar, das sich die Population im Laufe der Generationen gebildet hat. Über zwei Faktoren ist die Anpassungsgeschwindigkeit oder Mutabilität der Schrittweiten einstellbar. Die exogene Kontrolle bei der zweigliedrigen Strategie wurde also abgelöst durch eine *Selbstadaptation* (siehe Kapitel 5.2). Die mehrgliedrige Strategie lernt folglich auf zwei Ebenen, da eine schnelle optimale Einstellung der Objektvariablen ohne entsprechende Anpassung der Strategievariablen nicht möglich ist.

Das Verfahren, wie es bisher vorgestellt wurde, kann nur die Skalierung der Mutationen entlang den Koordinatenachsen im \mathbb{R}^n leisten, indem die σ_i -Relationen untereinander korrekt eingestellt und langsam angepaßt werden. Wie in Abbildung 2 gezeigt, erhält man die volle Drehbarkeit der Höhenlinien gleicher Mutationswahrscheinlichkeitsdichte erst dann, wenn man optional noch vererb- und mutierbare Lagewinkel in das genetische Material aufnimmt. Mathematisch gesprochen, werden auf diese Weise korrelierte Veränderungen der Variablen erzeugt. Die Konvergenzgeschwindigkeit erhöht sich bei passenden strategischen Parametern erheblich. Bei unpassender Schrittweitenwahl, z.B. wenn sich die Suche vorübergehend auf niedriger dimensionale Unterräume beschränkt, kann zwar kurzfristig auch eine höhere Konvergenzrate erzielt werden; dies wird aber langfristig mit einem Geschwindigkeitsverlust erkauft, weil die ‚Überanpassung‘ der Strategievariablen erst allmählich wieder abgebaut werden kann. Dieses Phänomen läßt sich durch geeignete Maßnahmen vermeiden (siehe Kapitel 5.2).

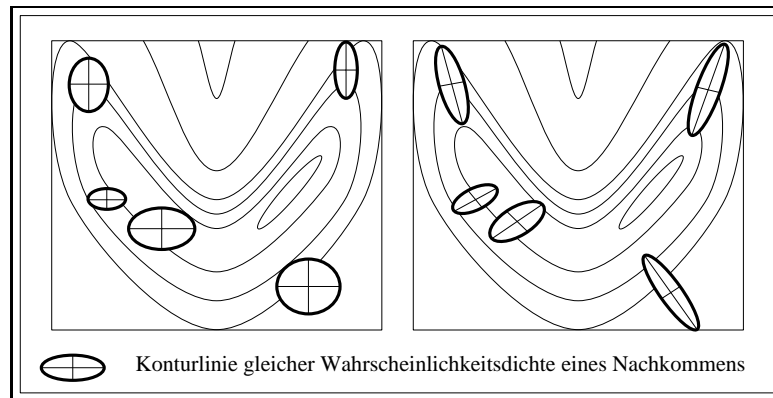


Abbildung 2: Unkorrelierte und korrelierte Mutationen (aus [13])

Für die genetische Kontrolle von Strategievariablen spricht einerseits die Existenz von Reparaturenzymen, andererseits die Beobachtung solcher Phänomene wie Polygenie bzw. Pleiotropie: mehrere Gene sind für die Ausprägung eines Merkmals verantwortlich bzw. ein Gen beeinflusst mehrere Merkmale. In Kapitel 5.2 werden die Auswirkungen dieses komplexeren internen Modells diskutiert.

Die μ Eltern erzeugen pro Generation λ Nachkommen in folgenden drei Schritten:

- Objekt- und Strategievariablen werden gemäß gewähltem Rekombinationstyp ‚zusammengewürfelt‘, wobei alle Eltern unabhängig von ihrer Fitneß über gleiche Chancen verfügen, ihre Gene zu vererben.
- Im zweiten Schritt mutieren zunächst die Strategievariablen des gerade erzeugten Nachkommen, deren neue Werte dann zur Mutation der Objektvariablen herangezogen werden.
- Im Falle einer Plus-Strategie überleben die besten μ aus den $\mu + \lambda$ Individuen, bei der Komma-Variante werden die μ Eltern der nächsten Generation nur aus den λ Nachkommen ausgewählt. Im ersten Fall kann ein Individuum also im Prinzip unendlich lange überleben, während im zweiten Fall seine Lebensdauer auf genau eine Generation begrenzt ist. Weitere Zwischenformen sind denkbar. Das ‚Vergessen‘ guter Lösungen erscheint auf den ersten Blick als Verschwendung von Rechenzeit, ermöglicht dem Verfahren aber eine problemlosere Anpassung der Strategieparameter und besonders bei multimodalen Problemen das Verlassen lokaler Optima. Ferner werden die oben erwähnten zwischenzeitlichen Überanpassungen schnell ausgemerzt. Besonders deutlich kann dies wie in Abbildung 3 bei einem über der Zeit wandernden Optimum veranschaulicht werden. Die Plus-Variante hält an den einmal gefundenen Lösungen fest, während die Komma-Strategie dem periodisch weiterspringenden Optimum folgen kann.

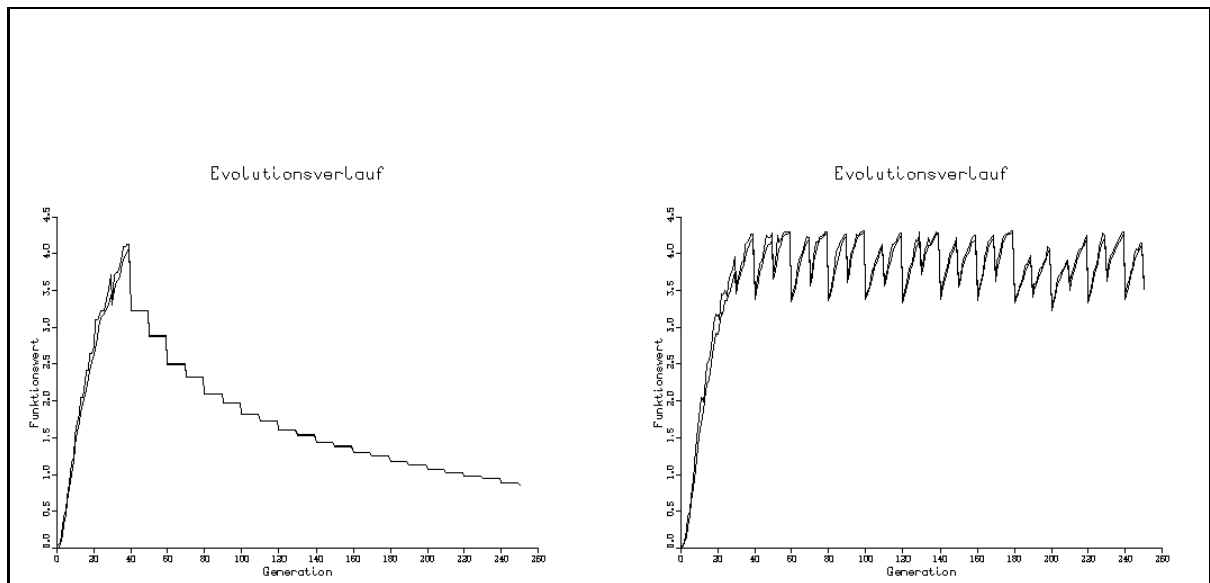


Abbildung 3: Plus- (links) und Komma-Strategie (rechts) bei wanderndem Optimum

Durch die Wahl des Verhältnisses μ/λ kann eingestellt werden, ob man durch einen kleinen Quotienten eine schnelle, aber nur lokale Konvergenz wünscht, oder ob man durch einen größeren Quotienten, durch eine ‚weichere‘ Selektion, eher die globalen Sucheigenschaften betonen möchte. Rekombination ist natürlich erst für $\mu > 1$ möglich.

Der Spezialfall einer $(1 + \lambda)$ -Evolutionstrategie konnte für dieselben Modellfunktionen wie bei der Herleitung der $1/5$ -Erfolgsregel analytisch durchleuchtet werden. Die berechneten Wahrscheinlichkeiten einer erfolgreichen Mutation liegen bei $1/\lambda \approx 1/6.0$ bzw. $1/\lambda \approx 1/4.7$, also relativ nahe bei $1/5$. Stellt man $\mu/\lambda \approx 1/6$ ein, wird man eine schnelle lokale Konvergenz des Verfahrens beobachten. Wird dieses Verhältnis Eins, irrt die (λ, λ) -Strategie nur noch ziellos durch den Parameterraum (‘Random Walk’).

Nicht nur diese Einstellmöglichkeit unterscheidet die zwei- von der mehrgliedrigen Evolutionstrategie. Weitere Unterschiede leiten sich vor allem daraus ab, daß die mehrgliedrige Strategie mit einer echten Population arbeitet, also eine Durchmischung individueller Merkmale durch Rekombination des genetischen Materials zweier oder auch mehrerer Individuen stattfinden kann. Dieser Austausch des verteilten Wissens erweist sich auch deshalb als sinnvoll und notwendig, da man nicht erwarten kann, daß ein Mitglied der Population *allein* über die vollständige, perfekte Information verfügt. Die Bedeutung der Rekombination für das Optimierverfahren wird in Kapitel 5.2 noch genauer dargestellt.

5.1 ... als universelles Optimierverfahren

Mit Hilfe von Evolutionstrategien lassen sich fast beliebige Optimierprobleme lösen, sofern man die Eingangsgrößen des betrachteten Systems identifizieren kann und dieses System ein Qualitätsmaß zurückgibt, so daß eine Selektion möglich ist. Es wird also nur ein Minimum an Information über die zu lösende Aufgabe benötigt — insbesondere keine expliziten Ableitungen und auch keine impliziten Annahmen über Stetigkeit bzw. Differenzierbarkeit. Somit können hochdimensionale, multimodale, nichtlineare, diskrete/stetige Probleme mit linearen und/oder nichtlinearen Restriktionen verarbeitet werden. So kann sich hinter dem Zielfunktionswert auch das Resultat eines Simulationslaufes verbergen, und Restriktionen können z.B. Festigkeitsforderungen repräsentieren, die mittels einer Finite-Element-Methode berechnet werden müssen [7]. Durch relativ geringe Modifikationen der mehrgliedrigen Evolutionstrategie sind inzwischen auch Probleme mit mehrfacher Zielsetzung (Vektoroptimierung) lösbar [20], ebenso wie NP-vollständige kombinatorische Probleme wie das des Handlungsreisenden [1, 23] — dies natürlich nur näherungsweise — oder binäre Aufgaben [25]. Auch für experimentelle Anwendungen ist dieses Verfahren geeignet: Der Rechner berechnet Vorschläge, und ein Mensch übernimmt die Selektion nach subjektiven, nicht oder nur schwer formalisierbaren Kriterien. So gibt es ein ‘Insektenzuchtprogramm’, wie es in [9] angeregt wurde. Die immer preiswerter werdende parallele Hardware kommt allen Strategien besonders zugute, die sich an biologischen Phänomenen orientieren (siehe Kapitel 5.2). Ein Konzept zur Nutzung der den Evolutionstrategien innewohnenden skalierbaren Parallelität existiert [14] und ist in [23] auch schon erfolgreich umgesetzt worden.

Evolutionstrategien haben ferner in einem umfangreichen Leistungsvergleich mit den gängigen mehrdimensionalen Optimierverfahren bewiesen, daß sie bezüglich der Konvergenzgeschwindigkeit zur Spitzengruppe gehören und daß die mehrgliedrige Strategie sogar alle anderen Verfahren schlägt, wenn es um die Konvergenzsicherheit geht, ohne aber eine Garantie für das Auffinden des globalen Optimums geben zu können. Aber es zeigte sich auch, daß

es bei linearen und quadratischen Zielfunktionen schnellere Verfahren gibt, nämlich genau jene, die für diese Spezialfälle entworfen wurden und daher über das perfekte interne Modell verfügen, das die Evolutionsstrategien sich erst bilden müssen. Der Vergleich wurde anhand von 50 unterschiedlich schwierigen Zielfunktionen durchgeführt [26].

5.2 ... als selbstorganisierendes Optimierverfahren

Analytische Resultate für die Konvergenzraten lassen sich einerseits nur für spezielle Zielfunktionen und andererseits nur für bestimmte Evolutionsstrategien herleiten. So betrachtete Rechenberg [21] die $(1 + 1)$ - und Schwefel [26] die $(1 + \lambda)$ -Variante. Für die interessanten Strategien mit $\mu > 1$ muß man sich bisher mit Simulationsresultaten begnügen. Im folgenden werden also Resultate solcher Computer-Experimente vorgestellt, die das Lernen einer geeigneten *Skalierung* und *Metrik* demonstrieren. Minimiert werden dafür die beiden folgenden Zielfunktionen, wobei es nicht so sehr auf das Resultat als vielmehr auf den Verlauf des Prozesses dorthin ankommt:

$$f_1(\underline{x}) := \sum_{i=1}^{30} (i \cdot x_i^2), \quad f_2(\underline{x}) := \sum_{i=1}^{10} \left(\sum_{j=1}^i x_j \right)^2$$

In den Grafiken wird als Fortschrittsmaß jeweils der Term

$$\log \sqrt{\frac{f^{(Start)}(\underline{x})}{f^{(aktuell)}(\underline{x})}}$$

benutzt.

Für den Test auf Erlernen einer Skalierung wurde f_1 verwendet. Fortlaufend sind also $n = 30$ Schrittweiten passend zueinander einzustellen, um schnell voranzukommen.

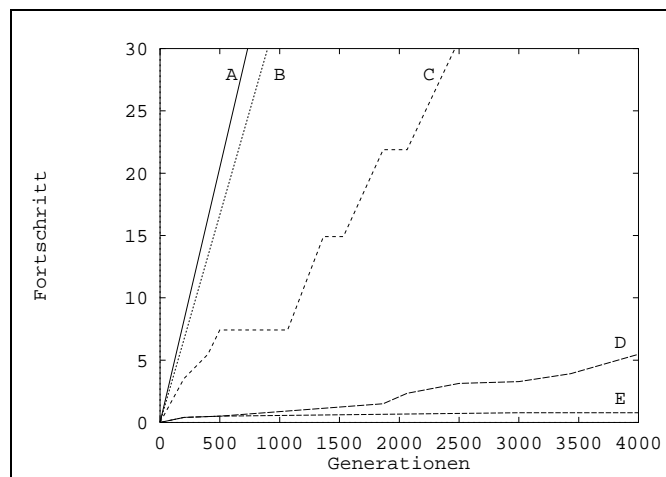


Abbildung 4: Erlernen einer Skalierung

Hinter der Kurve E in Abbildung 4 verbirgt sich eine $(1, 10)$ -Strategie, die deshalb versagt, weil die Konvergenzrate umgekehrt proportional zu n abnimmt (siehe auch Kapitel 4). Wenn nur der beste Nachkomme überleben darf, führt dies dazu, daß derjenige sich durchsetzt, der nur in einem Unterraum des \mathbb{R}^n operiert. Eine Vergrößerung von λ allein kann diese Überadaptation nicht verhindern. Erst die gleichzeitige Vergrößerung von μ und λ weist einen Ausweg aus dieser Sackgasse. Das Überleben-Lassen nur des ‚Besten‘ erweist sich also als Fehlinterpretation des Geschehens in der Natur. Hinter Kurve D verbirgt sich eine $(3, 6)$ -Strategie mit Rekombination. Bereits hier kann man den Effekt der zusätzlichen Prinzipien *Population* und damit *Rekombination* erkennen, was durch Kurve C mit einer $(6, 30)$ -Variante (mit Rekombination) noch weiter bestätigt wird.

Wie gut bei einer mehrgliedrigen Strategie mit Rekombination die Anpassung funktioniert, kann allerdings nur der Vergleich mit der bestmöglichen Variante zeigen. Kurve A zeigt die Geschwindigkeit einer $(1, 100)$ -Strategie, die bereits mit optimal voreingestellten Schrittweitenrelationen ($\sigma_i = c/\sqrt{i}$) startet und deshalb auf die Rekombination verzichten kann. Eine $(15, 100)$ -Evolutionstrategie mit Rekombination, die sich dieses Wissen erst im Lauf der Suche beschaffen muß (Kurve B), kommt nur unwesentlich langsamer voran. Die Kurven B bis E in Abbildung 4 zeigen also den Zuwachs an Fortschrittsgeschwindigkeit mit wachsender Population und zeigen eindrucksvoll, daß Lernen nur kollektiv stattfinden kann — eine gute Übereinstimmung zwischen Urbild Natur und ‚Modell‘ (siehe Kapitel 1).

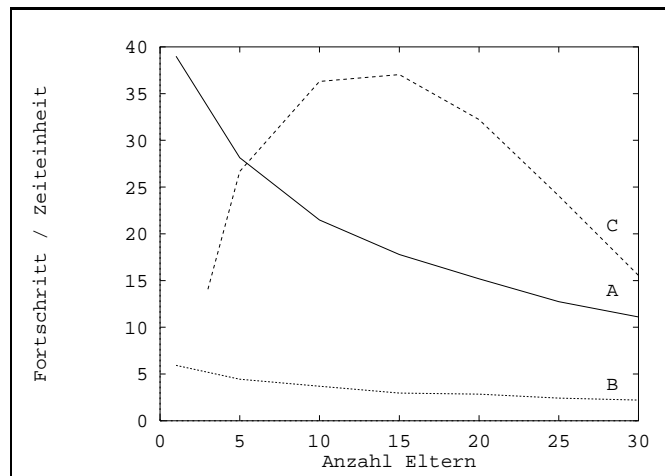


Abbildung 5: Vergleich der Konvergenzgeschwindigkeiten

In diesem Zusammenhang taucht die Frage nach dem *optimalen Selektionsdruck* auf: Wieviele der Nachkommen sollen zu Eltern der nächsten Generation werden? Die Antwort gibt Abbildung 5. Die Simulationsläufe wurden wieder mit f_1 durchgeführt, allerdings wurde nun μ bei konstantem $\lambda = 100$ variiert. Fall A repräsentiert wieder das Verfahren mit optimal

voreingestellten σ_i -Relationen, während für Kurve B zufällige, also nicht-optimale Schrittweitenrelationen eingestellt wurden. Kurve C konnte die Schrittweiten selbst anpassen.

In den Fällen, in denen es nur eine (Gesamt-)Schrittweite zu lernen galt (A und B), erwies sich die Wahl von $\mu = 1$ als die beste. Wenn nur ein σ auf der strategischen Parameterebene anzupassen ist, kommt man offenbar ohne Diversität der Eltern aus. Dagegen muß sich μ etwa zwischen 12 und 20 bewegen, wenn eine Skalierung erfolgreich gelernt werden soll. Auch hier zeigt sich wieder die Effektivität des kollektiven Lernens: Die (15, 100)-Evolutionstrategie konvergiert mit einem niedrigeren Selektionsdruck fast genauso schnell wie Variante A mit perfekter Skalierung und ist somit sogar schneller als eine (15, 100)-Variante mit richtig voreingestellten σ_i -Relationen. Dies kann man nur als *Synergieeffekt* interpretieren: 15 ‚Dumme‘ leisten als *Kollektiv* mehr als die gleiche Anzahl an ‚Spezialisten‘.

Schließlich soll anhand der Zielfunktion f_2 noch demonstriert werden, daß eine mehrgliedrige Strategie mit genügend weicher Selektion zusätzlich noch eine (positiv definite) Metrik erlernen kann, d.h. es sind nun neben den 10 Schrittweiten noch $n \cdot (n - 1) / 2 = 45$ Lagewinkel bzw. Kovarianzen einzustellen.

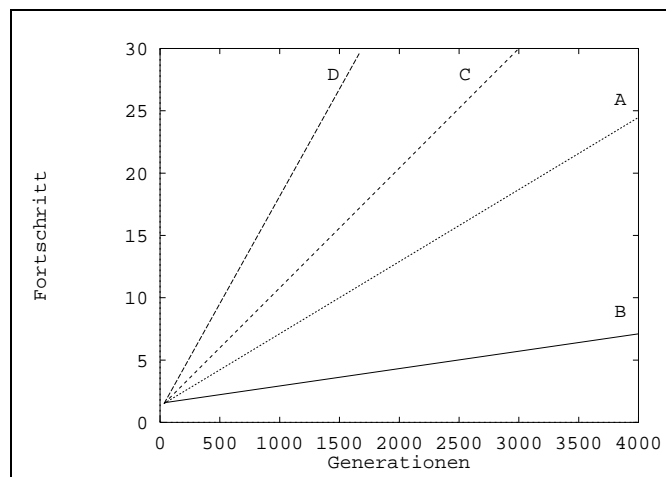


Abbildung 6: Erlernen einer Metrik

Die Kurven A und B in Abbildung 6 zeigen eine (1, 100)- bzw. (15, 100)-Evolutionstrategie mit fixierten, aber nicht optimalen Schrittweiten. Beide Verfahren arbeiten ohne Rekombination. Kurve C dagegen nutzt diese Möglichkeit, Wissen zwischen Individuen auszutauschen und kann auch die Schrittweiten einstellen. Die Strategie aber, die am schnellsten vorankommt (Kurve D), verfügt zusätzlich noch über die Möglichkeit, die Lagewinkel einzustellen. Der Suchprozeß kann also durch Einführung eines komplexeren inneren Modells bei entsprechender Zielfunktionstopologie gegenüber jenen Varianten noch beschleunigt werden, die nur eine Skalierung erlernen können.

Bisher zeigten die Abbildungen 4 und 6 nur den Fortschritt der Zielfunktionswerte über der Zeit. Genauer gesagt, wurde die durchschnittliche Fitneß der μ Eltern alle 50 Generationen dargestellt. Betrachtet man jedoch die Entwicklung der Werte einer Entscheidungsvariablen x_i allein, kann man noch weitere interessante Details erkennen (siehe Abbildung 7). Da die Zielfunktion von 30 Parametern abhängt, muß die Entwicklung einer Variablen nicht der des Suchprozesses als Ganzem entsprechen. Große Fortschritte einer Variablen in Richtung auf das Optimum erlauben zumindest bei nichtlinearen Zielfunktionen Rückschritte in anderen Richtungen. Besonders deutlich wird dies bei zu hohem Selektionsdruck, z.B. bei einer (5, 100)–Evolutionstrategie: Dann läuft der Evolutionsprozeß nicht mehr kontinuierlich, sondern in Sprüngen zwischen Stagnationsphasen ab. „Punctuated equilibria“ lautet der Begriff, den Biologen für die Beschreibung solch sprunghafter Entwicklungen verwenden [10], wenn sich Phasen großer Änderungen mit solchen relativer Ruhe periodisch abwechseln. Als Fortschrittsmaß wurde hier

$$\log \left| \frac{x_i^{(Start)}}{x_i^{(aktuell)}} \right| \quad \text{mit } i = 15$$

gewählt.

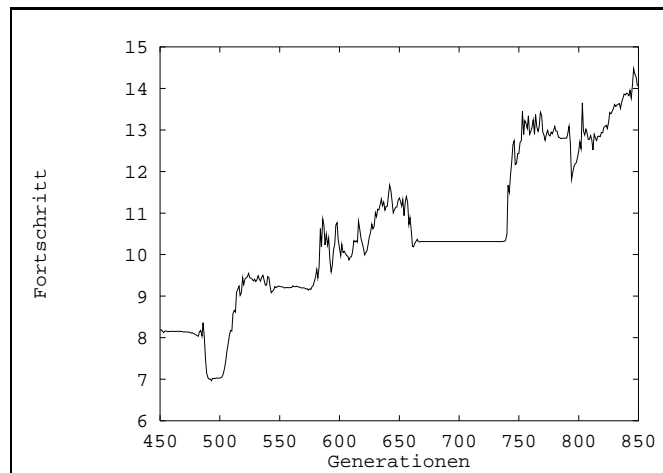


Abbildung 7: Entwicklung einer Entscheidungsvariable

Die Selbstorganisation bzw. das Selbstlernen in Evolutionsstrategien hängt von folgenden Faktoren ab, wobei noch einmal auf die gute Übereinstimmung zwischen den Erkenntnissen der Biologie und den Simulationen im Rechner verwiesen sei:

- Die Population muß zu jedem Zeitpunkt hinreichend groß sein, um eine genügend große Diversität zu gewährleisten. Nicht nur der momentan ‚Beste‘ einer Population darf sich also fortpflanzen, sondern eine Menge von ‚besseren‘ Individuen. In der Biologie spricht

man von der „requisite variety“, die nötig ist, damit eine Art nicht genetisch verarmt und schließlich ausstirbt.

- Innerhalb dieser Population müssen die Individuen ihr angesammeltes Wissen mit dem anderer Individuen rekombinieren.
- Die Bereitschaft, unter Umständen auch Rückschritte von einer Generation zur nächsten in Kauf zu nehmen, ist wichtig, damit sich passende interne Modelle für den schnellen weiteren Fortschritt durchsetzen können. Eine nur endliche Lebensdauer von Organismen drückt so gesehen keine Unvollkommenheit der Natur aus, sondern ist wichtig, um das genetische ‚Einfrieren‘ der jeweiligen Art zu verhindern.

Läßt man sich darauf ein, daß Evolutionsstrategien ein zumindest einfaches Modell natürlicher Prozesse darstellen, so kann man die Simulationsresultate auch nutzen, um diese Prozesse besser zu verstehen. Das Motto — warum nicht auch für den Alltag? — könnte dann lauten: „Gemeinsame Experimente statt rücksichtsloser Konkurrenz“ [28], um die auf allen Gebieten vorhandene Diversität auch nutzen zu können.

6 ABGRENZUNG ZU ANDEREN NATURANALOGEN, SELBSTORGANISIERENDEN VERFAHREN

Die nächsten ‚Verwandten‘ der Evolutionsstrategien, was Einsatzbereich, Grundidee und Zeitpunkt der Entwicklung betrifft, sind die von Holland [15] entwickelten *Genetischen Algorithmen*. Dennoch gibt es einige Unterschiede:

- Bei genetischen Algorithmen sind die zu optimierenden Parameter durch ein in der Regel binäres Alphabet kodiert, sie operieren also auf Bitstrings. Man könnte auch sagen, die ‚Black Box‘ verfügt nicht mehr über Regler, sondern über Ein/Aus-Schalter [11], was diese Algorithmen geeigneter für diskrete Aufgaben erscheinen läßt.
- Rekombination und Mutation sind in genetischen Algorithmen ebenfalls vorhanden, allerdings mit einer ganz anderen Rollenverteilung: Der wichtigste Suchoperator oder ‚Motor‘ ist hier die Rekombination, während die selten auftretenden Mutationen dafür Sorge tragen sollen, den Verlust einer Null oder Eins an einer bestimmten Bit-Position zu verhindern. Die Wahrscheinlichkeiten, mit denen diese beiden Operatoren auftreten, werden exogen vorgegeben und können daher nicht adaptiv angepaßt werden. Lernen findet also nur auf der Ebene der Objektvariablen statt.
- Die Selektion in genetischen Algorithmen wird anders als in Evolutionsstrategien vorgenommen: Es werden keine überschüssigen Nachkommen erzeugt, sondern ein Individuum erhält proportional zu seinem Anteil an der Gesamtfitneß der Population Platz auf einem ‚Glücksrad‘ zugeteilt (proportionale Selektion), das die Reproduktionswahrscheinlichkeiten der Individuen widerspiegelt. Ein besonders gutes Individuum kann dadurch sehr schnell die Population dominieren und die Suche sogar bei einfachen, unimodalen Zielfunktionen (siehe Abbildung 1 links) behindern. Durch andere Selektionsmechanismen (z.B. Ranking) läßt sich diese Schwäche mildern.

Ebenfalls Mitte der 60er Jahre entwickelte John von Neumann die Idee der *zellulären Automaten* [30]. Diese bringen mit wenigen einfachen, strikt lokalen Regeln komplexe globale Strukturen hervor. Das „Game of Life“ dürfte der bekannteste und einfachste Automat dieser Art sein. In ernsthaften Anwendungen kann man die Ausbreitung von Infektionskrankheiten, periodisch ablaufende chemische Reaktionen, Partikelkollisionen in Gasen und Flüssigkeiten und auch dendritische (baumähnliche) Kristallisationsprozesse simulieren, aber keine Optimierungsaufgaben lösen.

Noch ein Jahrzehnt früher beschrieben Metropolis et al. den Verlauf von thermodynamischen Abkühlprozessen. Für die Optimierung ist dies insofern von Bedeutung, als diese Prozesse eine Kristallstruktur auf minimalem Energieniveau erreichen, wenn sie genug Zeit zur Verfügung haben. Die Methode des *Simulated Annealing* kann als universelles Optimierverfahren angesehen werden [18], das wie folgt funktioniert:

- Ein Mechanismus sorgt für zufällige Änderungen der Systemzustände und bietet diese der Selektionskomponente an. Wenn man so will, bildet dieser Schritt das Analogon zur Mutation in den evolutionären Algorithmen.
- Eine Änderung, die in einem ‚besseren‘ Zustand resultiert, wird immer akzeptiert; die Wahrscheinlichkeit, mit der auch Verschlechterungen angenommen werden, hängt von einem Kontrollparameter — der Temperatur — ab, die, einem *Abkühlschema* folgend, nach einer gewissen Zahl von Zustandsänderungen reduziert wird, wodurch sich auch die oben erwähnte Wahrscheinlichkeit exponentiell reduziert. Für eine gegebene Aufgabe eine geeignete Anfangstemperatur und ein gutes Abkühlschema zu finden, erfordert jedoch viel Erfahrung oder einige Experimente.

Das bereits 1949 von dem Psychologen Hebb vorgeschlagene einfache Modell kindlichen Lernens gab aus heutiger Sicht den Startschuß für die (künstlichen) *Neuronalen Netze*, die sich an der Funktionsweise natürlicher Zentralnervensysteme orientieren. Obwohl dieses Gebiet heute wohl wegen der sich überschneidenden Aufgabenbereiche der Künstlichen Intelligenz zugerechnet wird, ist die ‚Intelligenz‘ dieser Netze nicht in der Form lokalisierbar wie in einzelnen Regeln eines Expertensystems, sie verteilt sich eher nach Art eines Hologramms über die ganze Struktur. Die folgende Tabelle faßt die heute verbreiteten Netzwerkmodelle und ihre Eigenschaften zusammen [4]:

<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

Ihr wahres Leistungsvermögen werden die Neuronale Netze aber erst entfalten können, wenn sie nicht mehr auf sequentiellen Rechnern simuliert werden müssen, sondern Netztopologien auf Hardware-Ebene dynamisch erzeugt und verändert werden können.

Fortschritte können aber auch schon erzielt werden, wenn die naturanalogen Verfahren untereinander rekombiniert werden: So können genetische Algorithmen genutzt werden, um die Topologie neuronaler Netze für eine Aufgabe zu optimieren (z.B. in [12, 19]). Oder die eigentlich feste Lern-Rate neuronaler Netze wird — wie die Temperatur beim Simulated Annealing — variiert, was einerseits die Konvergenzgeschwindigkeit gegenüber dem Backpropagation-Algorithmus erhöht und andererseits neue chaotische Attraktoren hervorbringt [29]. Ferner können Evolutionsstrategien das Gradientenverfahren ersetzen, das dem Backpropagation-Verfahren zugrundeliegt, um ein Optimierverfahren mit eher lokalen gegen eines mit globalen Eigenschaften auszutauschen [24]. Schließlich bestehen große Ähnlichkeiten zwischen Hopfield-Netzen und dem Verfahren des Simulated Annealing [16]. Naheliegender ist auch der Austausch zwischen den evolutionären Algorithmen. So konnten z.B. die genetischen Algorithmen in die Lage versetzt werden, die Mutationswahrscheinlichkeit bei veränderter Selektion während des Laufes selbst einzustellen [3].

7 AUSBLICK

Können nicht die in Kapitel 6 erwähnten Verfahren lediglich verschiedene Ansichten einer einzigen facettenreichen Münze ‚Natur‘ sein, wenn man sich auf die jeweils fundamentalen Ideen zurückzieht und von den Unterschieden im Detail abstrahiert?

- Den Verfahren werden nur wenige Parameter vorgegeben, das Phänomen ‚Lernen‘ bringen die sich selbst überlassene Verfahren hervor. Brooks [6] pointiert diesen Sachverhalt insofern, als er Intelligenz bei Systemen mit einem gewissen Komplexitätsgrad als ihnen innewohnendes, manchmal schwer zu ergründendes Phänomen konstatiert.
- Alle Algorithmen benutzen den Zufall, um bereits erreichte Zustände modifiziert der jeweiligen Bewertungs- oder Selektionskomponente vorzuschlagen.
- Eine selektive Stabilisierung ‚guter‘ Zustände sorgt für das äußerlich sichtbare ‚intelligente‘ Verhalten.
- Das Wissen um einen guten Zustand oder den Weg dorthin ist über viele ‚dumme‘ Elemente verteilt, die folglich ihre Fähigkeiten nur in *Kooperation* mit anderen ausspielen können.
- Im Gegensatz zu anderen künstlich intelligenten Systemen wie z.B. Expertensystemen gibt es bei den naturanalogen Verfahren keine zentrale Kontrolle. Dies bietet auf der einen Seite den Vorteil, daß man kein Wissen als Eingabe und damit als Startvoraussetzung benötigt, hat aber auf der anderen Seite den Nachteil, daß keine Erklärungskomponente Erfolge oder, wohl wichtiger, Mißerfolge verstehen helfen kann — hier kann ein Grund für die bisher geringe Verbreitung dieser Ideen liegen. So ist das Resultat einer großen Simulation [17] noch nicht in die Realität übertragen worden: Einem Rechner in einem Rechnernetz wird eine umfangreiche Aufgabe übertragen. Er darf sich aber gegen einen auszuhandelnden fiktiven Preis, den er aus seinem Startkapital

bezahlt, von den anderen helfen lassen. Eine gleichmäßige Auslastung des Netzes — dies war das Ziel — kann ohne zentrale Steuerung erreicht werden. Dies dauert länger, wenn in einem heterogenen Netz Rechner einer höheren Leistungsklasse (und entsprechend höheren Preisen) erst von anderen gemeinsam etwa gleichzeitig ‚entdeckt‘ werden müssen, um den Preis durch Teilen aufbringen zu können. Warum soll man nicht die Zeitspanne bis zur gleichmäßigen Auslastung aller dadurch verkürzen können, daß man die Standard-Rechner gleich mit dem Wissen starten läßt, wie Exoten einzubeziehen sind? Die Simulation dieser Idee sorgt für heftige Oszillationen und sogar Chaos im Netz, weil sich alle gleichzeitig sofort die leistungsfähigeren Rechner leisten können, diese völlig überlastet und dadurch uninteressant werden, worauf alles wieder von vorn beginnt.

Zufall, ‚gezähmt‘ durch Auslese- oder Rückkopplungsprozesse, verliert viel von seiner Bedrohlichkeit — erst er bietet einem System die Chance zur *Selbstorganisation*: Störungen können abgefangen und für die Entdeckung neuer stabiler Zustände genutzt werden, was man auch als intelligentes Verhalten interpretieren kann.

Diejenigen, die ein Unbehagen verspüren, wenn Verfahren Einzug in das Alltagsleben finden sollen, deren Verhalten sich nicht durch eine Zerlegung vollständig ergründen läßt, in denen der Zufall die treibende Kraft ist, mögen sich als Trost an den letzten weltweiten Börsencrash erinnern, als ein hochgradig geordnetes System mit starren Regeln ohne ein drastisches volkswirtschaftliches Signal plötzlich ins Chaotische umkippte.

LITERATUR

- [1] P. ABLAY, *Optimieren mit Evolutionsstrategien*, Spektrum der Wissenschaft, Juli 1987, S. 104–115.
- [2] W. R. ASHBY, *Design for a Brain (2nd Ed.)*, Wiley & Sons, New York, 1960.
- [3] T. BÄCK, *Self-Adaptation in Genetic Algorithms*, in: Proceedings of the First European Conference on Artificial Life, December 11-13, 1991, Paris, France, 1992, The MIT Press, S. 263–271.
- [4] T. BARR, *Netze im Aufwind*, c't, Apr. 1991.
- [5] 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.
- [6] R. A. BROOKS, *Intelligence without Reason*, MIT A.I. Memo No. 1293, Apr. 1991.
- [7] I. CAMPOS PINTO, *Wissensbasierte Unterstützung bei der Lösung von Optimierungsaufgaben*, Dissertation, Universität Dortmund, Fachbereich Informatik, 1989.
- [8] G. B. DANTZIG, *Lineare Programmierung und Erweiterungen*, Springer, Berlin, 1966.
- [9] R. DAWKINS, *Reise in das Land der Biomorphen*, Bild der Wissenschaft, Okt. 1987.
- [10] N. ELDREDGE UND S. J. GOULD, *Punctuated Equilibria: The Tempo and Mode of Evolution Reconsidered*, Paleobiology, Jahrgang 3, 1977, S. 115–151.

- [11] D. E. GOLDBERG, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison–Wesley, Reading, Massachusetts, 1989.
- [12] S. HARP, T. SAMAD UND A. GUHA, *Towards the Genetic Synthesis of Neural Networks*, in: Proceedings of the International Conference on Genetic Algorithms, J. D. Schaffer (Hrsg.), San Mateo, 1989, Morgan Kauffman, S. 360–369.
- [13] F. HOFFMEISTER UND T. BÄCK, *Genetic Algorithms and Evolution Strategies: Similarities and Differences*, in: Parallel Problem Solving from Nature, Proceedings of the 1st PPSN–Workshop, Dortmund, 1990, H.-P. Schwefel und R. Männer (Hrsg.), Jahrgang 496 *Lecture Notes in Computer Science*, Berlin, 1991, Springer, S. 445–469.
- [14] 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.
- [15] J. H. HOLLAND, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Michigan, 1975.
- [16] J. J. HOPFIELD, *Neural Networks and Physical Systems with Emergent Collective Computational Abilities*, in: Proc. Nat. Acad. Sci. USA, 1982, S. 2554–2558.
- [17] B. A. HUBERMAN, *The Ecology of Computation*, Studies in Computer Science and Artificial Intelligence, North–Holland, Amsterdam, 1988.
- [18] S. KIRKPATRICK, C. D. GELATT UND M. P. VECCHI, *Optimization by Simulated Annealing*, Science, Jahrgang 220, 1983, S. 671–680.
- [19] H. KITANO, *Designing Neural Networks Using Genetic Algorithms with Graph Generation Systems*, Complex Systems, Jahrgang 4, 1990, S. 461–476.
- [20] F. KURSAWE, *A Variant of Evolution Strategies for Vector Optimization*, in: Parallel Problem Solving from Nature — Proc. 1st Workshop PPSN, H.-P. Schwefel und R. Männer (Hrsg.), Jahrgang 496 *Lecture Notes in Computer Science*, Universität Dortmund, 1.–3. Oktober 1990, Springer, Berlin, 1991, S. 193–197.
- [21] I. RECHENBERG, *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Frommann–Holzboog, Stuttgart, 1973.
- [22] R. ROSEN, *Optimality Principles in Biology*, Butterworths, London, 1967.
- [23] G. RUDOLPH, *Global Optimization by means of Distributed Evolution Strategies*, in: Parallel Problem Solving from Nature, Proceedings of the 1st PPSN–Workshop, Dortmund, 1990, H.-P. Schwefel und R. Männer (Hrsg.), Jahrgang 496 *Lecture Notes in Computer Science*, Berlin, 1991, Springer, S. 209–213.
- [24] R. SALOMON, *Verbesserung konnektionistischer Lernverfahren, die nach der Gradientenmethode arbeiten*, Dissertation, TU Berlin, Fachbereich Informatik, 1991.
- [25] H.-P. SCHWEFEL, *Binäre Optimierung durch somatische Mutation*, may 1975.
- [26] ———, *Numerische Optimierung von Computer–Modellen mittels der Evolutionsstrategie*, Birkhäuser, Basel, 1977.

- [27] —, *Numerical Optimization of Computer Models*, Wiley & Sons, Chichester, 1981.
- [28] —, *Collective Phenomena in Evolutionary Systems*, in: 31st Annual Meeting of the International Society for General System Research, Budapest, 1987, S. 1025–1033.
- [29] P. F. M. J. VERSCHURE, *Chaos-based Learning*, Complex Systems, Jahrgang 5, 1991, S. 359–370.
- [30] J. VON NEUMANN, *Theory of Self-Reproducing Automata*, University of Illinois Press, 1966.