

# Grundlagen und Anwendungen Evolutionärer Algorithmen

## Abschlußbericht des BMBF Verbundprojektes EVOALG

Projektleiter: Prof. Dr. W. Ebeling  
Mitantragsteller: Prof. Dr. L. Schimansky-Geier,  
Dr. Dr. T. Pöschel  
Humboldt Universität zu Berlin

Wissenschaftliche Mitarbeiter:  
T. Aßelmeyer, D. Beule, V. Buchholtz,  
U. Erdmann, C. Kappler, R. Lieske, B. Militzer, A. Neiman,  
H. Rosé, D. Rosenkranz, F. Schautz,  
B. Tilch, M. Zamparelli

Berlin, den 29. Juni 1998



## **1.2 Kurze Darstellung: Aufgabenstellung, Voraussetzungen, Ablauf, Stand, Zusammenarbeit**

Der vorliegende Bericht enthält die Ergebnisse der Gruppe an der Humboldt Universität (Prof. Ebeling), die in vierjähriger Arbeit im Projekt EVOALG – Grundlagen und Anwendung evolutionärer Algorithmen – erhalten wurden. Dieses Projekt wurde als Verbundprojekt der Humboldt Universität, des Informatik-Zentrums Dortmund (Leiter: Prof. Dr. H.-P. Schwefel) und der Siemens AG (Gruppe Dr. M. Höhfeld) vom Bundesministerium für Forschung und Technologie (FKZ 01 IB 403 B3) gefördert.

Die Gruppe an der Humboldt Universität setzte sich überwiegend aus theoretischen Physikern zusammen. Es gabe umfangreiche Vorkenntnisse und Erfahrungen auf dem Gebiet der Statistischen Physik und nichtlinearen Dynamik, bei der Modellierung von Evolutionsprozessen und weiterhin einige Arbeiten zu Grundlagen von Optimierungsprozessen. Die Aufgabe bestnad darin, mit modernen Methoden der statistischen Physik, in Zusammenarbeit mit den Kooperationspartnern, neue evolutionäre Algorithmen zu entwickeln und auf konkrete Probleme des Partners (Siemens AG) sowie andere Fragen von praktischen Interesse anzuwenden.

Ausgehend vom Stand und der Aufgabenstellung war der Schwerpunkt der Arbeit auf die Untersuchung von praxisrelevanten Modellproblemen, die Entwicklung optimaler Optimierungsalgorithmen, die Klassifikation von Optimierungsproblemen, sowie die theoretische Seite der Anwendungsprobleme, die gemeinsam mit den Verbundpartnern bearbeitet wurden, gerichtet.

Im Rahmen des Projektes wurde auch die internationale Konferenz PPSN – Parallel Problem Solving from Nature“, die im September 1996 in Berlin stattfand, mitorganisiert, insbesondere betrifft das die Begutachtung der Beiträge und die Gestaltung des Programms.

Die Logik unseres Berichtes folgt dem Arbeitsplan und den wichtigsten Ergebnissen, die erzielt wurden. Der Aufbau ist wie folgt:

Im ersten Kapitel werden Methoden zur Strukturanalyse von Sequenzen (Gensequenzen, Buchstabenfolgen etc.) vorgestellt. Diese Arbeiten sind nach unserer Auffassung von Interesse für Anwendungen in der Bioinformatik, aber auch im Hinblick auf die Entwicklung optimaler Kompressionsalgorithmen interessant.

Der Schwerpunkt des zweiten Kapitels ist die Klassifikation von Optimierungsproblemen. Folgende Aufgabe stand hier im Vordergrund: Am Beginn einer Optimierung ist es in der Regel sehr nützlich, schon möglichst viel über die allgemeinen Eigenschaften des Problems zu wissen, soweit sich das mit einfachen Hilfsmitteln herausfinden läßt. Die von uns neu entwickelten Methoden erlauben es, eine Entscheidung darüber zu treffen, welcher Typ von Optimierungsalgorithmen für das gegebene Problem besonders geeignet ist, ob ein bestimmter Mutationsoperator einem anderen vorzuziehen ist, etc..

Im dritten Kapitel werden im Detail verschiedene neue, bzw. verbesserte Optimierungsalgorithmen vorgestellt. Es handelt sich dabei zum Beispiel um gemischte Strategien, in denen die biologisch orientierten evolutionären Algorithmen mit thermodynamisch orientierten Boltzmannalgorithmen (Simulated Annealing) dem jeweiligen Problem entsprechend kombiniert werden. Weitere Ansätze bilden einerseits die Methode der adaptiven Suchräume, die ein effizientes Verfahren zur Lösung ganzzahlig-gemischter Optimierungsprobleme darstellt und andererseits die Entwicklung robuster Mutationsoperatoren für die Parameteroptimierung, die sowohl globale, als auch lokale Suche zulassen.

Im vierten Kapitel wird die Anwendung der entwickelten Algorithmen auf praxisrelevante Modellprobleme der strukturellen Optimierung dargestellt. Es handelt sich dabei u.a. um das wichtige Problem der Faltung von Biopolymeren, das eine der zentralen Fragen im Design neuer Wirkstoffe und Medikamente ist; um das LABS – Least autocorrelated Binary Sequences – Problem, das in der Nachrichtentechnik seine Bedeutung besitzt und Probleme der Netzwerkoptimierung, die sowohl in der Städte- und Verkehrsplanung, als auch in der Kommunikationstechnik derzeit zunehmend an Bedeutung gewinnen.

Im fünften und letzten Kapitel werden einige Anwendungen evolutionärer Algorithmen auf Probleme der Physik vorgestellt. Ein Schwerpunkt der Arbeit war das von der KWU über die Gruppe der Siemens AG in Auftrag

gegebene Problem der Optimierung des Einsatzes von Brennstäben in Kernkraftwerken. Dieses Problem, wurde in intensiver Zusammenarbeit mit den Projektpartnern bearbeitet. Bearbeitet wurde besonders die theoretische Seite des Problems, weiterhin wurden verschiedene Beiträge zur Lösung der praktischen Optimierungsprobleme gegeben. Details dazu sind im Abschlußbericht der Kooperationspartner zu finden. Unser theoretischer und methodischer Beitrag wurde in Publikationen niedergelegt, die unter Punkt 3. zu finden sind. Andere Probleme mit Praxisrelevanz, die von uns bearbeitet wurden, ist das Problem der Sekundärstruktur von Biopolymeren (RNA) sowie Probleme der Optimierung von Netzwerken.

Die folgenden Kapitel enthalten eine eingehende Darstellung der erzielten Ergebnisse gegliedert nach Schwerpunkten der Arbeit.

W. Ebeling

Berlin, den 29. Juni 1998



# Inhaltsverzeichnis

<b>1</b>	<b>Strukturanalyse von Sequenzen</b>	<b>9</b>
1.1	Aufgabenstellung und Analysemethoden . . . . .	9
1.2	Statistische Sequenzanalyse . . . . .	10
1.3	Die Symbolische Dynamik und Referenzsysteme . . . . .	11
1.4	Optimierung von Sequenzdarstellungen - Kompressibilität . . . . .	12
1.5	Langreichweitige Korrelationen in Texten . . . . .	12
1.6	Erstellte Software . . . . .	13
1.6.1	Programm SEQENTRO . . . . .	13
1.6.2	Programm BLOCK . . . . .	13
1.6.3	Programm SEQZIPF . . . . .	14
1.6.4	Programm TRANSINF . . . . .	14
<b>2</b>	<b>Klassifikation von Problem und Strategie</b>	<b>15</b>
2.1	Das Grundmodell . . . . .	15
2.2	Die Genotyp-Phänotyp Abbildung als glättende Repräsentation . . . . .	17
2.3	Klassifikation von Optimierungsproblemen . . . . .	21
2.3.1	Die Zustandsdichte . . . . .	21
2.3.2	Das LABS Problem I . . . . .	28
2.3.3	Die Zustandsdichte von Multi-Layer-Perceptronen . . . . .	30
2.3.4	Eine Methode zum Vergleich von Mutationsoperatoren . . . . .	35
2.4	Klassifikation von Optimierungsstrategien . . . . .	35
2.5	Erstellte Software . . . . .	44
2.5.1	Seq . . . . .	44
<b>3</b>	<b>Optimale Algorithmen und adaptive Strategien</b>	<b>47</b>
3.1	Die Boltzmann Strategie . . . . .	47
3.2	Die Darwin Strategie . . . . .	50
3.3	Gemischte Strategien . . . . .	54
3.4	Die adaptive Darwin Strategie . . . . .	56
3.5	Selfannealing . . . . .	62
3.6	Die Anwendung der Methode der adaptiven Suchräume . . . . .	64

3.6.1	Zusammenfassung . . . . .	64
3.6.2	Adaptive evolutionäre Optimierung – Konvergenzbeschleunigung durch Erweiterung des Zustandsraumes . . . . .	66
3.7	Robuste Mutationsverteilungen in kontinuierlichen Suchräumen	74
3.8	Die analytische Formulierung der $(\mu, \lambda)$ -Strategie . . . . .	80
3.9	Optimierung mittels Brownscher Teilchen . . . . .	80
3.10	Erstellte Software . . . . .	83
3.10.1	PDE2D . . . . .	83
3.10.2	ArtEvo . . . . .	85
3.10.3	Mix . . . . .	86
3.10.4	XMix . . . . .	87
3.10.5	SAS . . . . .	88
3.10.6	MMix . . . . .	89
3.10.7	XBola . . . . .	90
3.10.8	Fema . . . . .	91
3.10.9	XFema . . . . .	92
3.10.10	ProEvo . . . . .	93
3.10.11	Das Programm evo . . . . .	94
<b>4</b>	<b>Strukturelle Optimierung</b>	<b>107</b>
4.1	RNA Sekundärstrukturen . . . . .	107
4.2	Das LABS-Problem II . . . . .	110
4.3	Optimierung von Straßennetzen . . . . .	111
4.4	Adaptive Netzwerkoptimierung . . . . .	112
4.5	Erstellte Software . . . . .	113
4.5.1	Wea . . . . .	113
<b>5</b>	<b>Anwendungen auf Probleme der Physik</b>	<b>121</b>
5.1	Das Beladeplanproblem von Druckwasserreaktoren . . . . .	121
5.2	Molekulardynamik mit Cauchyverteilungen . . . . .	121
5.3	Die Anwendung evolutionärer Algorithmen auf quantenmechanische Probleme . . . . .	122

# Kapitel 1

## Strukturanalyse von Sequenzen

### 1.1 Aufgabenstellung und Analysemethoden

(W. Ebeling)

Sequenzen von Buchstaben spielen eine zentrale Rolle in der modernen Wissenschaft und Technik. Die meisten Informationsträger wie z.B. Texte, Aufzeichnungen auf Disks und Tapes, DNA- und Protein-Moleküle haben in Bezug auf ihre Primärstruktur die Form linearer Buchstabenfolgen. Auch viele Optimierungsprobleme können als Sequenzen dargestellt werden, so z.B. das Problem des Reisenden Handelsmanns (TSP) und das Problem minimaler Autokorrelationsfunktionen (LABS). Aus diesem Grunde wurde der Analyse der Struktur und der Darstellungen von Sequenzen ein spezielles Thema im Projekt gewidmet. Obwohl es sich dabei (bis auf die Frage optimaler Darstellungen) nicht unmittelbar um Optimierungsfragen handelt, spielten die Resultate dieses Teilprojektes eine wichtige Rolle in allen übrigen Themen. Das Thema Sequenzanalyse stand am Beginn der Forschungen für das Projekt EVOALG und wurde in den wesentlichen Teilen schon relativ früh abgeschlossen. Dabei sind auch eine größere Zahl wissenschaftlicher Publikationen (siehe Anlage) sowie zwei Dissertationen (A. Schmitt [Sch94], J. Freund [Fre95]) und mehrere Diplomarbeiten entstanden. Darüber hinaus zeigte eine Reihe von Einladungen zu Kolloquiumsvorträgen in der BRD (u.a. Aachen, Bayreuth, Dresden, Potsdam, Rostock) sowie Vorträgen auf Konferenzen das allgemeine wissenschaftliche Interesse an diesem Themenkreis. Das wichtigste Resultat der Arbeit an diesem Thema ist die Bereitstellung eines ganzen Spektrums von Methoden für die Analyse der Struktur von Sequenzen [EP94, HES94, EPA95, EN95, ENP95, PER95, EPA96]. Dazu zählen:

- (1) Entropien höherer Ordnung (Blockentropien) [EP94, EPA95],
- (2) bedingte bzw. dynamische Entropien [EPA95],
- (3) Transinformationen (mutual information) [HES94, EPA95],
- (4) algorithmische Entropien und Kompressibilitäten [ENP95, EPA96],
- (5) Fluktuationsanalyse der lokalen Zusammensetzung [EN95, ENP95]
- (6) charakteristische Exponenten für Random-Walk-Darstellungen [ENP95],
- (7) Fourierspektren [EN95].

Die erarbeiteten Methoden sind inzwischen im Innovationskolleg “Theoretische Biologie” der HUB bereits sehr erfolgreich für die Analyse von DNA- und Protein-Sequenzen eingesetzt worden (AG Herzel). In letzter Zeit ergaben sich eine Reihe neuer Gesichtspunkte: Insbesondere konnte gezeigt werden, daß Fragen der Prognose von Zeitreihen und komplexen nichtlinearen Prozessen mit unseren Methoden der Entropieanalyse erfolgversprechend bearbeitet werden können. Die Nutzung dieser Methoden soll in Kooperation mit dem Innovationskolleg der HUB, der Universität Brüssel, der Universität Potsdam (Prof. Kurths), dem Potsdamer Institut für Klimafolgenforschung (Prof. Schellnhuber) und dem Institut für ökologische Probleme (BIT”OK) der Universität Bayreuth (Prof. Hauhs) verfolgt werden. Ein weiteres Anwendungsfeld ergibt sich bei Optimierung von Kompressionsalgorithmen. Die von uns berechneten Entropien geben Schranken für die Kompression von Sequenzen an. Wir konnten anhand verschiedener Tests zeigen (Resultate von A. Neiman), daß die soweit verwendeten Standard-Algorithmen wie z.B. gzip, rar, arj, x1 noch ziemlich weit vom theoretischen Optimum entfernt liegen [EPA96]. Bei der Anwendung von Kompressionsalgorithmen auf Sequenzen mit bekannter Struktur (vgl. dritter Abschnitt) schneidet gzip in der Regel am besten ab. Die Frage der Optimierung von Sequenzen wird weiter unten noch einmal im Abschnitt LABS angesprochen.

## 1.2 Statistische Sequenzanalyse

(W. Ebeling)

Die im Rahmen des Themas vorgenommenen statistischen Analysen von Sequenzen dienen in erster Linie der zuverlässigen Berechnung von Block-Entropien [EPA95]. Dabei entstand das Problem der Bestimmung einer

großen Zahl von Wahrscheinlichkeiten aus einem relativ kleinen Daten-Material. Dazu wurden umfangreiche Untersuchungen von Ebeling, Herzel, Pöschel, Rosé, Schmitt u.a. angestellt, die zur Ausarbeitung zuverlässiger Methoden für die Schätzung von Wahrscheinlichkeiten, Entropien, Transinformationen usw. sowie zu einer größeren Zahl von Publikationen in anerkannten internationalen Journalen führten (siehe Publikationsliste und besonders die Arbeiten [HES94, EPA95, PER95]). Bearbeitet wurden im Detail:

- (1) Biosequenzen [HES94],
- (2) Textsequenzen [EP94, EPA95]
- (3) durch nichtlineare Dynamik generierte Sequenzen [FERck]
- (4) durch grammatische Regeln generierte Sequenzen [RFE96],
- (5) Sequenzen mit optimierten globalen Eigenschaften (z.B. LABS).

### 1.3 Die Symbolische Dynamik und Referenzsysteme

(W. Ebeling)

Im Rahmen des Teilthemas Sequenzanalysen war es zunächst notwendig, eine Reihe von Standard-Sequenzen mit bekannten Eigenschaften bereitzustellen. Dieses Vorgehen erwies sich als sehr fruchtbar und führte zu einer Reihe wertvoller Resultate. Insbesondere konnten in Arbeiten von Ebeling, Freund und Rateitschak die statistischen Eigenschaften von verschiedenen nichtlinearen Abbildungen, besonders der logistischen Abbildung und intermittenter Prozesse analytisch behandelt werden [EFR96, FERck, REF96, RFE96]. Diese analytisch behandelten Sequenzen wurden als Testsysteme für die Ausarbeitung von Methoden der Sequenzanalyse eingesetzt. Diese Forschungen führten ebenfalls zu einer Reihe von Publikationen [EFR96, FERck, REF96, RFE96]. und Konferenzbeiträgen (u.a. Gordon-Konferenz 1994). Frau K. Rateitschak erhielt inzwischen ein EU-Stipendium und führt ihre Untersuchungen bei Prof. G. Nicolis an der Universität Brüssel fort.

## 1.4 Optimierung von Sequenzdarstellungen - Kompressibilität

(W. Ebeling)

Die Darstellung von Sequenzen und die Optimierung der Darstellung ist ein Problem von zentraler Bedeutung für die moderne Informatik (Darstellung von Texten, Programmen, Datenmassiven usw.). Ein wichtiges Anwendungsfeld ergibt sich bei der Optimierung von Kompressionsalgorithmen. Die von uns berechneten Entropien geben Schranken für die Kompression von Sequenzen an. Wir konnten anhand verschiedener Tests zeigen (Resultate von A. Neiman), daß die soweit verwendeten Standard-Algorithmen wie z.B. gzip, arj, rar noch ziemlich weit vom theoretischen Optimum entfernt liegen. [EPA96]. Dabei wurden insbesondere die im 3. Abschnitt besprochenen Testsequenzen mit bekannten Eigenschaften eingesetzt. Es konnte gezeigt werden, daß die standardmäßig verwendeten Kompressionsalgorithmen nur einen Teil der vorhandenen Regularitäten bzw. grammatischen Strukturen aufdecken können. Diese Resultate wurden u.a. auf der Konferenz Coherent Structures 1995 in Kyoto [ENP95] und auf der PhysComp 1996 in Boston [EPA96] vorgestellt. Die Frage der Optimierung von Sequenzen wird weiter unten noch einmal im Abschnitt LABS angesprochen.

## 1.5 Langreichweitige Korrelationen in Texten

(A. Neiman)

Im Zusammenhang mit der Frage optimaler Kompressionsalgorithmen wurden auch langreichweitige Korrelationen in Texten untersucht. Die zugrundeliegende Hypothese ist, daß die Kompressibilität einer Sequenz ein Maß für die in ihr steckenden Korrelationen ist. Ein Ziel war es, Korrelationen zu finden, die von den Standard-Kompressionsalgorithmen noch nicht ausgenutzt werden, und somit Raum für Verbesserungen eröffnen.

Langreichweitige Korrelationen können mithilfe von dynamischen Entropien, Korrelationsfunktionen und Standardabweichungen,  $1/f^\delta$  Rauschen, kritischen Exponenten, Kumulanten höherer Ordnung, sowie Transinformationen gemessen werden. Wir gehen davon aus [EN92, AEN94], daß ein Text strukturelle Ähnlichkeit mit symbolischen Sequenzen hat, die durch nicht-lineare Prozesse am Bifurkationspunkt erzeugt wurden. Die Korrelationen solcher symbolischer Sequenzen sind weitgehend bekannt. Unsere Analyse [EN95] basiert auf einer Abbildung von Text auf Random-Walk Modelle

[Pen92] sowie einer Spektralanalyse. Außerdem untersuchen wir die Kompressibilität der symbolischen Sequenzen.[ENP96, EPA96]

Auf der Suche nach der Ursache langreichweitigen Korrelationen haben wir Texte auf verschiedenen Ebenen umgeordnet bzw. geschüttelt (Buchstaben, Wörtern, Sätzen, Kapiteln usw.). Die Korrelationen des umgeordneten Textes wurden mit denen des Originaltexts verglichen. Korrelationen auf höheren Ebenen, die durch größere Zusammenhänge im Text zustande kommen, wurden nur im Originaltext erhalten. Alle Originaltexte zeigen anomale Diffusion, d. h. langreichweitige Korrelationen, während die umgeordneten/geschüttelten Texte normale Diffusion zeigen.

Unsere Untersuchungen zeigen, daß die Skalierung der Standardabweichung (gemessen in Hölderexponenten), Spektralanalyse, Momente höherer Ordnung, sowie Kumulanten geeignete Maße für langreichweitige Korrelationen in Texten, jenseits der Satzebene, sind. Es wurde gezeigt, daß langreichweitigen Korrelationen in Texten ihre Wurzel in der Struktur des Textes (Worte, Sätze, Kapitel usw.) haben. Die Ergebnisse sind von Interesse für die Analyse von Kompressionsalgorithmen und die Abschätzung vorhandener Reserven.

## 1.6 Erstellte Software

(W. Ebeling)

### 1.6.1 Programm SEQENTRO

SEQENTRO ist ein Programm zur Berechnung der Entropien höherer Ordnung (Blockentropien) von Sequenzen, die aus Buchstaben auf einem Alphabet (mit  $\lambda$  verschiedenen Buchstaben) bestehen. Die Blockentropien  $H_n$  hängen von einer Laufzahl  $n$  ab.  $h_n = H_n + 1 - H_n$  sind die bedingten Entropien, die ein Maximum für die Vorhersagbarkeit darstellen. Das Programm läuft unter DOS/WINDOWS FORTRAN-Compilern. Es wurde mit Erfolg zur Berechnung der Struktur von Sequenzen eingesetzt, deren Länge  $10^5$  Buchstaben nicht überschreitet.

### 1.6.2 Programm BLOCK

BLOCK ist in Fortran geschrieben und besonders für Workstations geeignet. Beschreibung von BLOCK:

- (1) Es können Blockentropien bis zur Sequenzlänge  $10^7$  berechnet werden.
- (2) Die ranggeordneten Häufigkeiten werden angegeben
- (3) BLOCK besteht aus den files
 

-block.f	the program
-block.par	the basic parameters
-block.sequences	the batch file containing the names of the sequences and the description of the calculation to be done
-block.start	startup file, compilation
-block.doc	description

Ein komplettes demo-Paket liegt bei. Mit dem Befehl “demo” werden 4 demo-Sequenzen (Bernoulli) (dna1000.seq,...,dna1000000) erzeugt, die dann mit dem vorbereiteten file “block.sequences” verarbeitet werden. Dazu starte einfach “block.start” und man sieht, was passiert und welche files entstehen. Das Programm ist sehr schnell. Die Zipf-files lassen sich vorteilhaft mit gnuplot anschauen.

### 1.6.3 Programm SEQZIPF

SEQZIPF ist ein in Fortran verfaßtes Programm zur Berechnung der ranggeordneten Häufigkeiten von Wörtern der gegebenen Länge  $n$  für Sequenzen die aus Buchstaben auf einem Alphabet (mit  $\lambda$  verschiedenen Buchstaben) bestehen. Es folgt eine sogenannte Zipf-Verteilung, ähnlich wie in BLOCK. SEQZIPF ist für kürzere Sequenzen geeignet und wurde vorwiegend aus WINDOWS/DOS-Rechnern eingesetzt.

### 1.6.4 Programm TRANSINF

TRANSINF ist ein Programm zur Berechnung der Transinformation (englisch transinformation, mutual information) von Sequenzen die aus Buchstaben auf einem Alphabet (mit  $\lambda$  verschiedenen Buchstaben) bestehen. Die Transinformation hängt von einer Laufzahl  $n$  ab.  $I(n)$  ist ein Maß für die Stärke von Korrelationen im Abstand  $n$ . Die vorliegende Version läuft unter DOS/WINDOWS FORTRAN-Compilern. Das Programm wurde zur Berechnung der Struktur von Sequenzen bis zur Länge  $10^5$  eingesetzt.

# Kapitel 2

## Klassifikation von Problem und Strategie

### 2.1 Das Grundmodell

(H. Rosé)

Ziel der ersten Arbeitsphase war es, an einem Grundmodell, das einfach und handhabbar ist und dennoch die wichtigsten Eigenschaften realer Optimierungsprobleme aufweist, grundlegende Methoden der Untersuchung, Verbesserung und Anwendung von evolutionären Algorithmen zu etablieren. Die dabei gewonnenen Ergebnisse bildeten dann in der zweiten Phase die Basis zur erfolgreichen Konstruktion verbesserter Algorithmen.

Prinzipiell stellt ein Optimierungsproblem immer die Aufgabe dar, aus einer Vielzahl möglicher Lösungen, in angemessener Zeit die qualitativ hochwertigsten Lösungen zu gewinnen. Es ist damit definiert durch den Raum aller möglichen Lösungen oder Zustände - dem Suchraum - und durch eine Qualitätsfunktion - der Fitneß - die die Güte der Lösungen bewertet. Ein Optimierungsalgorithmus steuert eine Bewegung durch den Suchraum, die möglichst schnell die Optima der Fitneßfunktion aufzufinden vermag.

Suchräume können diskret (z.B. Menge von Buchstabenfolgen, von Netzwerken oder Anordnungen), kontinuierlich (z.B. Steuer- und Regelungsparameter) oder gemischt diskret-kontinuierlich sein. Schon hier zeigen Optimierungsprobleme eine deutliche Aufteilung im Schwierigkeitsgrad: Während kontinuierliche Suchräume günstige Eigenschaften in Bezug auf die Optimierung aufweisen, wie die stetige Veränderbarkeit der Suchparameter und damit die Möglichkeit der Schrittweitensteuerung der Suche oder die Angebarkeit des Abstands zweier Lösungen, besitzen diskrete Suchräume eine kompli-

zierte Struktur, die eine zielgerichtete und stete Optimierung erschwert oder sogar unmöglich macht, wenn keine geeignete Problemrepräsentation gefunden werden konnte.

Zu diesen den Suchraum bedingenden Fragen tritt als entscheidendes Kriterium einer erfolgreichen Optimierung der Komplexitätsgrad der Fitneßfunktion selbst. Diese ist fast immer nur algorithmisch gegeben und so der direkten Analyse unzugänglich. Multimodalität (Existenz vieler lokaler Optima), Frustriertheit (Existenz sich widersprechender Anforderungen) und Nichtstetigkeit (große Qualitätsänderungen bei kleiner Änderung der Lösung) bestimmen den Schwierigkeitsgrad der Optimierung.

All diese Faktoren gilt es zu berücksichtigen und zu verstehen, soll die Entwicklung anwendungsreifer und effektiver Optimierungsalgorithmen erfolgreich sein. Dabei spielen die Wahl einer geeigneten Problemrepräsentation und die Klassifikation des Schwierigkeitsgrades eine entscheidende Rolle. Dies berücksichtigend wurden im Projekt sowohl Struktur und Repräsentation diskreter Probleme als auch die Klassifikation von Fitneßlandschaften untersucht und auf die Konstruktion verbesserter Algorithmen angewendet.

Das erste Modell diskreter Optimierung war ein einfaches Evolutionsspiel zur Erzeugung frustrierter Sequenzen nach ENGEL [EF82, ERS94]. Eine frustrierte Fitneßfunktion setzt sich aus zwei oder mehr sich widersprechenden Optimierungskriterien zusammen. Die Lösungen des Optimierungsprozesses können dabei nicht beide Kriterien gleich gut erfüllen - die optimale Lösung stellt einen Kompromiß dar. Sowohl in der Natur wie in der Technik sind die Mehrzahl der Optimierungsaufgaben frustriert. Ein Beispiel dafür ist Leistungssteigerung einer technischen Apparatur bei gleichzeitiger Kostensenkung oder das im Projekt bearbeitete Beladeplanproblem für Kernreaktoren. Diese Systeme besitzen optimale Lösungen komplizierter Struktur.

Der Suchraum bestand aus Sequenzen des Alphabets mit  $\lambda$  Buchstaben und folgende Fitneßfunktion angewandt:

1. Jeder Buchstabe der Sequenz der mit seinem Nachfolger alphabetisch ist wird mit 1 bewertet.
2. Jeder Buchstabe der Sequenz der gleich seinem  $p$  ten Nachfolger ist wird mit  $b$  bewertet.

Bei  $\lambda \neq p$  liegt eine frustrierte Bewertung vor.

Zur Simulation des Evolutionsprozesses wurde eine Darwinstrategie mit Punktmutation und Turnierselektion betrachtet und in C implementiert (s.

2.5). Trotz der Einfachheit des Problems besitzen die optimierten Sequenzen eine auf langen Skalen nichtverschwindende Transinformation und sind damit von komplexer Struktur. Dies ließ es sinnvoll erscheinen, an diesem Modell die Fragen der Repräsentation und Klassifikation eingehender zu untersuchen.

## 2.2 Die Genotyp-Phänotyp Abbildung als glättende Repräsentation

(H. Rosé)

Es ist eine Tatsache, daß der Erfolg oder Mißerfolg der Optimierung eines Problems von seiner Repräsentation abhängt. Diese wird von der Kodierung des Problems und der Wahl der Mutationsoperatoren des evolutionären Algorithmus bestimmt. Bei geeigneter Wahl der Repräsentation kann ein vordem höchst kompliziertes Problem einfach und leicht optimierbar werden [AER96]. Allerdings stellt sich uns bei einem konkreten Problem sogleich auch die Frage: Was heißt *geeignete* Wahl? Beantwortbar und praxisrelevant wird diese Frage nur dann, wenn wir verstehen warum die Repräsentation im Stande ist das Problem zu vereinfachen.

Nun scheint die Beantwortung dieser Fragen recht aussichtslos bei der Vielzahl von möglichen Problemen und Optimierungsalgorithmen, wenn wir nicht schon ein Beispiel einer Optimierungsstrategie besäßen, die Antworten auf diese Fragen bereits fand: die biologische Evolution. Betrachten wir also dieses erfolgreiche Vorbild genauer.

Zuerst stellen wir fest, daß die biologische Evolution niemals die elementaren Objekte der Repräsentation - die Genotypen - bewertet, sondern daß immer erst den Phänotypen - den Eigenschaften der Individuen - eine Fitneß im Selektionsprozeß zugeordnet wird. Ein Individuum entsteht durch die Expression seiner DNA und besitzt dann Eigenschaften, die sein Überleben sichern. Es findet eine Abbildung der Gensequenz in den Raum der Eigenschaften statt.

Wenn wir diesen Prozeß etwas abstrahieren können wir sagen: Der Genotyp  $s$  als Element eines diskreten Sequenzraumes  $G$  wird auf einen Phänotyp  $\phi$  als Element eines kontinuierlich Eigenschaftsraumes  $Q$  abgebildet. Die Abbildung  $\Phi : s \in G \rightarrow \phi \in Q$  bezeichnen wir als *Genotyp-Phänotyp Abbildung*. Die Fitneßfunktion ist ein Ausdruck der Überlebensfähigkeit des im Selektionsprozeß stehenden Individuums; sie weist seinem Phänotyp eine reelwertige Qualität zu  $F : Q \rightarrow \mathbb{R}$ . Der Phänotyp  $\phi$  wird jedoch, durch Umkodierung

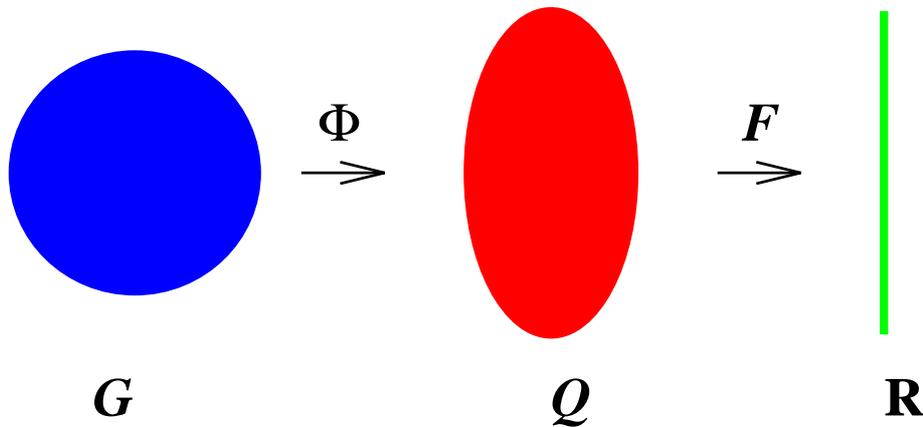


Abbildung 2.1: Genotyp-Phänotyp Abbildung

mittels  $\Phi$ , aus dem Genotyp  $s$  erzeugt. Also existiert eine Bewertungsfunktion  $V(s) = F(\Phi(s))$ , die auch dem Genotyp einen Fitneßwert zuordnet (s. Abb. 2.1). Das Optimierungsproblem kann also entweder durch die Valuationslandschaft  $V(s)$  über dem Genotypraum oder durch die Fitneßlandschaft  $F(\phi)$  über dem Phänotypraum repräsentiert werden. Die Genotyp-Phänotyp Abbildung beschreibt den Übergang zwischen beiden Repräsentationen, die in Bezug auf die Optimierung mehr oder weniger geeignet sein können.

Was bedeutet nun aber "geeignet"? Es ist wohl bekannt, daß eine erfolgreiche Optimierung das Prinzip der *starken Kausalität* erfüllen muß: kleine Änderungen der Lösungen müssen kleine Änderungen der Fitneß bewirken (RECHENBERG), d.h., daß die Fitneßlandschaft einen hohen *Grad an Stetigkeit* besitzen muß. Wenn also die Genotyp-Phänotyp Abbildung so gewählt werden kann, daß sich beim Übergang von  $V(s)$  zu  $F(\phi)$  der Grad an Stetigkeit erhöht, so erhalten wir aus der ursprünglichen genotypischen Repräsentation eine geeignetere phänotypische Repräsentation des Problems. Abb. 2.2 verdeutlicht dieses Resultat: Die Valuationslandschaft  $V(s)$  der Ge-

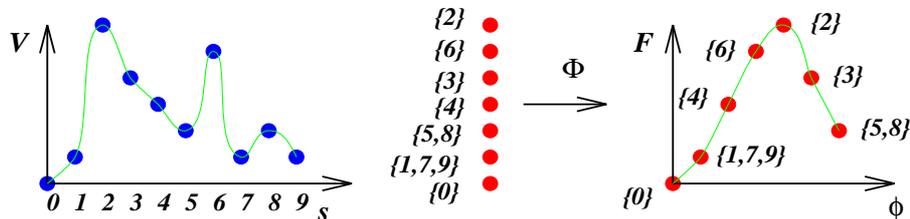


Abbildung 2.2: Glättende Wirkung der Genotyp-Phänotyp Abbildung

notypen  $s$  besitzt lokale Maxima und große Sprünge. Ganzen Klassen von

Genotypen ist der selbe Fitneßwert zu geordnet (z.B.  $\{1, 7, 9\}$ ,  $\{5, 8\}$ ). Zur Problemlösung ist es ausreichend nur einen Vertreter der Klasse zu finden. Wählen wir als Phänotypen genau diese Äquivalenzklassen von Genotypen gleicher Fitneß und ordnen wir diese so an, daß sich die Fitneß nur minimal ändert (z.B.  $\phi_0 = \{0\}$ ,  $\phi_1 = \{1, 7, 9\}$ ... $\phi_6 = \{5, 8\}$ ), so besitzt die neue Repräsentation  $F(\phi)$  nicht nur den Vorteil eines kleineren Suchraumes (10 Genotypen  $\rightarrow$  7 Phänotypen), sondern auch einen höheren Grad an Stetigkeit. Wir konnten im Ergebnis des Projektes zeigen, daß eine solche glättende Genotyp-Phänotyp Abbildung existiert ([AER96], Theorem 1).

Natürlich ist die genaue Kenntnis einer solchen Abbildung umfassender als die Lösung des Optimierungsproblems und so kaum angebar. Die Anwendbarkeit des erzielten Resultates liegt vielmehr in der Ausnutzung des Prinzips der Äquivalenzklassenbildung zur sukzessiven Konstruktion günstigerer Repräsentationen. Äquivalenzklassen korrespondieren zu dem Symmetrieeigenschaften des Problems. Die Ausnutzung dieser Beziehung bildet ein allgemeines Verfahren der Einbeziehung von problemspezifischen Wissen.

Dies sei an zwei konkreten Beispielen - dem Problem der ENGEL-Sequenzen (s. 2.1) und dem Beladepplanproblem (s. 5.1) - erläutert.

Beim ENGEL-Problem mit vier Buchstaben (**a,b,c,d**) können wir schrittweise folgende Repräsentationen konstruieren:

- Klasse zyklisch austauschbarer Buchstaben  
 Symmetrie: Fitneß gleich bei zyklischem Austausch:  $a \rightarrow b, \dots, d \rightarrow a$   
 Kodierung:  $\{a, b, c, d\} \rightarrow \{\text{@}, \#, \$, *\}$   
 Sequenz  $S \rightarrow$  Schema  $f$

$$f_i = \begin{cases} \text{@} & : \text{alphabetisch} \\ \# & : \text{periodisch} \\ \$ & : \text{beides} \\ * & : \text{keines} \end{cases} \quad F(f_i) = \begin{cases} 1 \\ b \\ 1+b \\ 0 \end{cases}$$

Beispiel: 
$$\begin{array}{l} S = \text{bcdadaa} \\ f = \text{@@@*\$\#\text{@}} \end{array}$$

- Klasse permutierbarer Schemata  
 Symmetrie: Fitneß unabhängig von Anordnung der  $\{\text{@}, \#, \$, *\}$   
 Kodierung:  $\text{@@@*\$ \#\text{@}} \rightarrow (n_1, n_2, n_3, n_4) = (1, 1, 4, 1)$ : Anzahl der ( $\$, \#, \text{@}, *$ )  
 Schema  $\rightarrow$  Schemavektor

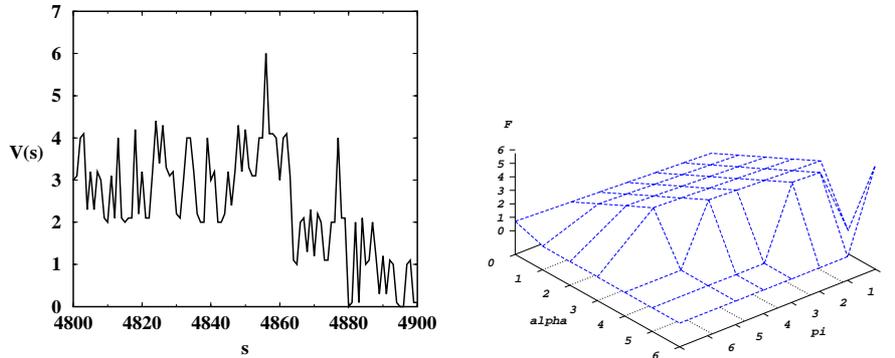


Abbildung 2.3: Valuationslandschaft der Sequenzen und Fitnesslandschaft der Phänotypen

$F$	Schema	\$	#	@	*
6.0	@@@@* bcdabca	0	0	6	1
5.2	*\$\$@* babcdaa	2	0	3	2
	@@*\$\$ bcdadaa	1	1	4	1

- Klasse der Fitnessniveaus  
 Symmetrie: Fitness nur abhängig von den Summen  $\alpha = n_1 + n_3$ ,  $\pi = n_1 + n_2$   
 Kodierung:  $(1, 1, 4, 1) \rightarrow (\alpha, \pi) = (5, 2)$ : Anzahl der (alphabetischen, periodischen) Buchstaben

$$\text{Schemavektor} \rightarrow \text{Phänotyp}$$

Mit jeder neuen Klasse reduziert sich die Größe des Suchraumes. Der Übergang von Sequenzen der Länge 7 zu den Phänotypen  $(\alpha, \pi)$  verkleinert die Größe um das Tausendfache. Die glättende Wirkung dieses Prozesses ist in Abb. 2.3 dargestellt.

Nicht nur die Wahl einer geeigneten Repräsentation auch die Verwendung effektiver Mutationsoperatoren beeinflussen die Effizienz des evolutionären Algorithmus. Ein Mutationsoperator der so konstruiert ist, daß er nur Genotypen in einander überführt die benachbarten Phänotypen angehören, Fitnesswert realisiert den Glättungsprozeß, ohne das die Genotyp-Phänotyp Abbildung explizit bekannt sein muß. Insofern sind die Konstruktion der Genotyp-Phänotyp Abbildung und die eines solchen Operators nur verschiedene Sichtweisen des selben Prinzips. Im Hinblick auf die Anwendbarkeit jedoch ist das

Einbeziehen des problemspezifischen Wissens in den Mutationsoperator vorteilhaft.

Im Falle des Beladeplanproblems eines Kernkraftwerkes (s. 5.1) sollten die optimalen Lösungen eine symmetrische Geometrie aufweisen, da der Abbrandprozeß sich diffusiv ausbreitet. Diese Symmetrie kann in einem Operator Berücksichtigung finden, der symmetrische Beladungen wieder in symmetrische überführt. Das ist äquivalent zur Wahl von Phänotypen, die als Klassen symmetrischer Beladungen gewählt werden. Ein solcher symmetrischer Operator wurde im Projekt konstruiert und es konnte gezeigt werden (s. 5.1), daß die Klasse der symmetrischen Beladungen bessere besitzt als die der asymmetrischen.

## 2.3 Klassifikation von Optimierungsproblemen

(H. Rosé)

Ein entscheidender Schritt zur Lösung eines Problems ist die richtige Einschätzung seiner Schwierigkeit. Sie bestimmt Wahl und Einsatz der Mittel. Wodurch jedoch wird die Schwierigkeit eines Optimierungsproblems charakterisiert? Können wir Klassen von ähnlich schweren Problemen angeben und für jede eine optimale Lösungsstrategie etablieren? Die Wichtigkeit der Beantwortung dieser Fragen liegt auf der Hand: Sollte eine Klassifikation der Probleme gelingen, könnten wenige optimale Lösungsstrategien konstruiert werden, die das heutige mühsame Probieren in zielgerichtete Anwendung überführen könnten. Aus diesen praktischen, aber auch theoretisch notwendigen Erwägungen wurde im Projekt die Frage der Problemklassifikation bearbeitet.

### 2.3.1 Die Zustandsdichte

Ein Optimierungsproblem ist durch seine Fitneßfunktion definiert. Diese kann man sich als Landschaft über dem Suchraum aller möglichen Lösungen oder Zustände vorstellen. Optimierung bedeutet nun das Auffinden der höchsten Berge dieser Landschaft. Das genaue Aussehen der Fitneßlandschaft eines konkreten Problems ist oft nicht bekannt und auch nicht von besonderer großer theoretischer oder praktischer Bedeutung, da nicht die Details, sondern die prinzipielle Struktur der Landschaft die Schwierigkeitsklasse eines Problems festlegt. Relevante Suchräume besitzen meist eine astro-

nomische Größe<sup>1</sup>. All dies macht die angestrebte Klassifikation zu keinem leichten Problem.

Fragen wir also zunächst nur das Einfachste: *Wieviele Lösungen existieren zu einem bestimmten Fitneßwert, insbesondere zum optimalen Wert?*

Diese Frage wird von einer Funktion beantwortet, die in der statistischen Physik wohl bekannt ist: die *Zustandsdichte*. Sie gibt die Anzahl der Zustände (Lösungen)  $n(F)$  in einem kleinen Fitneßintervall  $[F, F + \Delta F]$  an oder anders gesagt die Wahrscheinlichkeit einen bestimmten Fitneßwert zufällig zu finden. Außerdem ist sie invariant gegenüber der speziellen Darstellung des Problems, sie wird allein von der inneren Struktur der Landschaft bestimmt. Eben dies fordern wir aber von einem Klassifikationsmaß: es muß allein von der Struktur des Problems abhängen. Die Zustandsdichte wurde im Rahmen des Projektes als ein erstes Klassifikationsmaß für Optimierungsproblem definiert und an verschiedenen Beispielen untersucht [REA96].

Um einen analytischen Ausdruck für die Zustandsdichte zu bekommen, betrachten wir die nach oben und unten begrenzte Fitneßfunktion  $F_{min} < F(s) < F_{max}$  über dem Suchraum  $B$  im Intervall  $[F - \Delta F, F + \Delta F]$ . Dann erhalten wir für die Zustandsdichte  $n(F)$

$$n(F) = \lim_{\Delta F \rightarrow 0} \frac{\text{vol}(F^{-1}([F - \Delta F, F + \Delta F]))}{\text{vol}([F - \Delta F, F + \Delta F])} = \frac{d}{dy} \text{vol}(F^{-1}([F_{min}, y])) \quad (2.1)$$

Als ein Beispiel betrachten wir die Fitneßfunktion  $F : \mathbb{R}^n \rightarrow \mathbb{R}$  über dem  $n$ -dimensionalen Raum  $\mathbb{R}^n$  mit  $F(x_1, \dots, x_n) = x_1^2 + \dots + x_n^2$ . Die Zustandsdichte ist einfach durch

$$n(y) = \frac{d}{dy} \text{vol}(F^{-1}([0, y])) = \frac{n\pi^{n/2}}{(\frac{n}{2} + 1)} y^{n/2-1} \quad (2.2)$$

gegeben. Bild 2.4 zeigt den Verlauf dieser Funktion für die verschiedenen Dimensionen 5, 10 und 20 (die volle Linie ist die Dimension 5, die gestrichelten Linien sind die Dimensionen 10 bzw. 20). Die Schwierigkeit bei der Berechnung besteht in der Untersuchung der Menge  $\text{vol}(F^{-1}([0, y]))$ . Diese Menge besitzt die Struktur einer Mannigfaltigkeit, die schon in den einfachsten Fällen recht kompliziert ist.

Die meisten praktischen Probleme sind aber diskreter Natur, d.h. der Suchraum  $B$  besitzt eine diskrete Struktur. Bevor wir auf die Simulation der Zustandsdichten in solchen Räumen eingehen, wollen wir noch kurz die Unterschiede und die Gemeinsamkeiten zwischen dem kontinuierlichen

---

<sup>1</sup>So besitzt das Problem der Netzwerkoptimierung von 39 Punkten ca.  $10^{200}$  mögliche Lösungen, während die geschätzte Zahl von Elementarteilchen im Kosmos nur  $10^{80}$  beträgt.

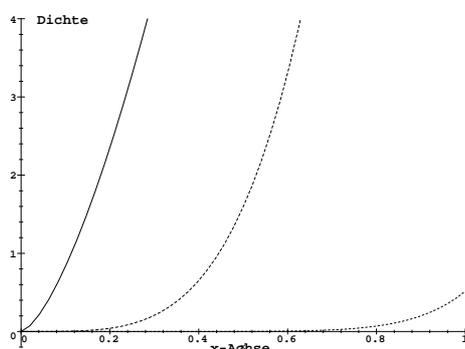


Abbildung 2.4: Zustandsdichte für Parabel in Abhängigkeit von der Dimension

und diskreten Fall anhand von einfachen Beispielen aufführen. Das erste Beispiel ist durch die Fitneßfunktion  $F : \mathbb{Z}^n \rightarrow \mathbb{Z}$  mit demselben quadratischen Ausdruck wie oben gegeben. Dann besteht das Problem darin, alle Lösungen von  $F(m_1, \dots, m_n) = k$  mit  $k, m_1, \dots, m_n \in \mathbb{Z}$  zu finden. Dies ist ein bekanntes Problem der Zahlentheorie. Entsprechend einem Satz von Siegel (siehe [HM73] Ch II, Th. 9.3) läßt sich für große  $n$  die Zustandsdichte auf den Fall (2.2) mit  $k$  anstatt  $y$  reduzieren. Als zweites Beispiel betrachten wir die Adjazenzmatrix  $A$  eines gerichteten Graphen. Wir geben jeder Verbindung  $A_{ij}$  zwischen  $i$  und  $j$  einen Wert durch die Beziehung  $(A_{ij}i \cdot j) = F(1, \dots, n) = G$ , wobei  $n$  die Anzahl der Verbindungen ist. Weiterhin seien zwei Matrizen äquivalent, wenn sie auf denselben Graph führen. Die gewichtete Zustandsdichte  $n(G)$  in Bezug auf die Äquivalenzrelation ist gegeben durch den Ausdruck von Siegel ([HM73], Ch. II, Th. 9.7)

$$n(G) \approx 0.705 \left( \frac{x}{2\pi e \sqrt{e}} \right)^{x^2/4} \left( \frac{8\pi e}{x} \right)^{x/4} / \sqrt[24]{x}$$

mit  $x$  als die Anzahl der Verbindungen.

Neben diesen konstruierten Beispielen, wurde für das Problem der ENGEL Sequenzen (siehe 2.1) die Zustandsdichte theoretisch und durch Simulationen bestimmt, wobei sich beide Ergebnisse decken. Der Wert des theoretischen Resultates besteht darin, die Beziehung zwischen Fitneßfunktion und Zustandsdichte aufzudecken. Dazu betrachten wir das Alphabet  $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$  mit 4 Buchstaben. Der Raum der Sequenzen wird mit  $G$  bezeichnet und die Abbildung  $V : G \rightarrow \mathbb{R}$  bewertet jede Sequenz  $s \in G$  nach den folgenden Regeln:

24 KAPITEL 2. KLASSIFIKATION VON PROBLEM UND STRATEGIE

1. Falls ein Buchstabe mit dem folgenden in alphabetischer Ordnung steht, dann setzen wir  $V = V + 1$ .
2. Falls ein Buchstabe gleich seinem  $p$ -ten Nachfolger ( $p = 5$  ab jetzt) ist, dann gilt  $V = V + b$  mit  $b < 1$ .

Bezeichnet  $S_i$  den  $i$ -ten Buchstaben der Sequenz  $s \in G$  dann gilt für die Bewertungsfunktion  $V$

$$V(s) = \sum_{i=1}^L [\alpha(S_i) + b\pi(S_i)] \quad (2.3)$$

wobei  $\alpha(S_i) = 1$  oder  $\pi(S_i) = 1$  bei Einhaltung der Regel 1 bzw. 2 gilt. Andererseits setzen wir einfach  $\alpha(S_i) = \pi(S_i) = 0$ . Der entscheidende Gedanke, der die Berechnung der Zustandsdichte erst möglich macht, besteht darin, daß alle Änderungen der Fitneß (Mutationen) eine Gruppe bilden, die mit  $\mathcal{G}$  bezeichnet wird. Einzelheiten kann man im Projekt entstandenen Artikel [AER96] nachlesen. Die Zustandsdichte  $n(V(s))$  ist dabei definiert durch

$$n(V(s)) = \sum_{st=gs, g \in \bar{\mathcal{G}}} \delta(V(st) - V(s)), \quad (2.4)$$

wobei  $\delta$  die Kroneckerfunktion ist und  $\bar{\mathcal{G}}$  diejenigen Gruppenelemente kennzeichnet, die alle Sequenzen durch Mutationen aus  $s \in G$  erzeugen. Die Ableitung läßt sich formal durch

$$\frac{\partial V}{\partial g}(s) = \frac{1}{\Delta}(V(st) - V(s)), \quad \Delta = \frac{1}{L} \quad (2.5)$$

mit  $L$  als Sequenzlänge einführen. Es sei bemerkt, daß man mit diesen Ableitungen genauso rechnen kann, wie mit den gewöhnlichen Ableitungen. Die Idee für die weitere Rechnung besteht darin, daß man die Ableitung

$$\frac{dn}{dV} = \sum_{g \in \mathcal{G}} \frac{\partial g}{\partial V} \frac{\partial n}{\partial g}. \quad (2.6)$$

bestimmt. Dabei spielt die Ableitung  $\partial g / \partial V$  eine große Rolle. Diese Ableitung beschreibt die Anzahl der Fitneßänderungen in Abhängigkeit von der Fitneß  $V$ . Eine einfache Argumentation mit Hilfe der Mutationsgruppe  $\mathcal{G}$  zeigt, daß

$$\frac{\partial g}{\partial V}(V) = A(V_0 - V) \quad (2.7)$$

für geeignete Konstanten  $V_0, A$  gelten muß. Damit erhält man die Differentialgleichung für die Zustandsdichte

$$\frac{dn}{dV} = -A(V - V_0)n \quad (2.8)$$

mit der Lösung für große Sequenzlängen  $L$

$$n(V) = n(0) \exp(-A(V^2 - V_0V)) \quad (2.9)$$

$$= n(0) \exp(AV_0^2/4) \exp(-A(V - V_0/2)^2). \quad (2.10)$$

Die noch offenen Konstanten lassen sich durch diese qualitative Rechnung nicht bestimmen, da sie in Abhängigkeit von der Sequenzlänge stark variieren können. Für die Rechnung wurde nur benutzt, daß die Fitneßfunktion (2.3) additiv ist und daß die Mutationen bis zu einem Wert  $V_0$  unabhängig sind. Jede solche Fitneßfunktion muß eine solche Gaußverteilung als Zustandsdichte haben.

Welche anwendbaren Aussagen kann man auf der Basis der Zustandsdichte treffen?

- Sie gibt die Wahrscheinlichkeit an, eine Lösung bestimmter Qualität durch zufällige Suche zu finden. Nun realisieren evolutionäre Algorithmen keine blinde zufällige Suche, sondern eine Erzeugung von Möglichkeiten bei anschließender Bewertung und besuchen damit nicht alle Bereiche des Suchraumes, sondern nur die momentan am erfolgversprechendsten. Insofern ist die Zustandsdichte nicht identisch mit der Wahrscheinlichkeit, daß ein evolutionärer Algorithmus einen gewissen Fitneßwert erreicht. Andererseits kann auch der beste Algorithmus nur das finden, was mit einer hinreichenden Wahrscheinlichkeit existiert. Im Ergebnis kann man also sagen:

Die Zustandsdichte limitiert die Wahrscheinlichkeit eine gute Lösung zu finden nach unten hin. Gerade ihr Verhalten im Abschnitt großer Fitneßwerte ist für die Beurteilung des Schwierigkeitsgrades entscheidend.

- Für alle relevanten Probleme wird die Zustandsdichte mit steigender Fitneß monoton fallen – die Anzahl immer besserer Lösungen sinkt stetig. Ihre Ableitung  $n'(H)$  in diesem Bereich bestimmt die Geschwindigkeit mit der sich die Chance verringert, eine noch bessere Lösung als bisher zu finden. Diese Tatsache ist von hoher praktischer Relevanz: Kenne ich die Zustandsdichte des Problems ist es mir möglich, eine Aufwands- und Nutzenabschätzung aufzustellen. Niemand wird weiterhin sein Rechenzeitkonto belasten wenn er abschätzen kann, daß die Zahl der z.B. um ein Promille besseren Lösungen um das Zehntausendfache kleiner ist, als die Zahl der Lösungen, die die gegenwärtig erreichte Qualität bereits realisieren.

Der Grad der Ableitung der Zustandsdichte ist ein direktes Maß der Schwierigkeit des Optimierungsproblems. Je höher der Grad ihres Abfalls desto schwieriger ist das Problem.

- Nimmt die Zustandsdichte mit steigender Fitneß immer weiter ab, so erreicht sie für das Optimum den Wert Null. Damit ist eine Abschätzung der maximal erreichbaren Qualität eines Optimierungsproblems möglich. Natürlich ist ihre Bestimmung genau in diesem Bereich stark approximativ, denn eine genaue Bestimmung würde ja die Kenntnis der Optima implizieren. Dieses Faktum ist jedoch nur für die theoretische Betrachtung problematisch. Relevante praktische Probleme fallen in die Klasse hoher Schwierigkeit. Das Absinken der Zustandsdichte ist bei ihnen besonders stark. Das bedeutet aber, daß die Zustandsdichte in einen sehr schmalen Fitneßintervall schlagartig Null wird. Die Lage dieses Intervalls kann als Abschätzung der maximal erreichbaren Fitneß angesehen werden.

Die Zustandsdichte schwerer Probleme ermöglicht eine Einschätzung und Erfolgswertung der bis dahin der erzielten Ergebnisse des Optimierungsalgorithmus.

Mit diesen Eigenschaften erweist sich das entwickelte Konzept der Zustandsdichte als wichtiges Ergebnis im Verständnis von Optimierungsproblemen. Dies würde jedoch wenig nutzen, wenn wir keine Möglichkeit haben die Zustandsdichte im konkreten Fall zu bestimmen. Unter Ausnutzung der Kenntnis der im Projekt untersuchten Boltzmannstrategie (s. 3.1) wurde eine Methode zur approximativen Bestimmung der Zustandsdichte für beliebige Probleme entwickelt und an Beispielen getestet [REA96].

Die offensichtlichste und damit auch wenig realistische Methode der Bestimmung ist das komplette Durchzählen des Suchraumes. Wie aussichtslos dieses Unterfangen ist wird deutlich wenn man versucht, die Zustandsdichte des ersten Beispiels (Netzwerkoptimierung, 39 Knoten) so zu bestimmen: Selbst wenn uns ein Parallelrechner zu Verfügung stände, bestehend aus sämtlichen Teilchen des Universums und jedes besäße eine Rechenleistung von eintausend GigaFlop, dann würde die komplette Suche  $10^{90}$  mal länger dauern als die Zeitspanne, die dem Alter des heutige Universum zugeschrieben wird.

Auch eine zufällige gleichverteilte Suche stößt an die Grenze der unvorstellbaren Größe des Suchraumes, da sie sich über den ganzen Suchraum verstreut.

Eine evolutionäre Dynamik besucht hingegen nur sehr kleine Teile des gesamten Raumes, sie konzentriert die Suche auf Bereiche die zur steten Verbesserung der Fitneß führen, alle anderen werden nicht betrachtet. Wenn man nun das Bewegungsgesetz dieser Dynamik kennt und aus ihm die letztendlich angestrebte stationäre Wahrscheinlichkeitsverteilung der Fitneßwerte

$P(F)$  einer Population von Suchern berechnen kann, so aus ihr die Bestimmung der Zustandsdichte möglich. Im Falle der Boltzmannstrategie kennt man die stationäre Verteilung – es ist die Boltzmannverteilung der statistischen Physik

$$P(F) \sim n(F) e^{-F/T},$$

und damit wird die Zustandsdichte bis auf Normierung durch  $P(F)$  festgelegt

$$n(F) \sim P(F) e^{F/T}.$$

Zur approximativen Bestimmung von  $P(F)$  simulieren wir ein Ensemble von Boltzmannsuchern, dessen zeitabhängige Wahrscheinlichkeitsverteilung für lange Zeiten gegen die  $P(F)$  konvergiert. Der folgende einfache Algorithmus realisiert das beschriebene Verfahren:

$x, y$	Zustnde
$F$	Fitness
$\varepsilon$	Niveau $F_\varepsilon$
$P[\varepsilon]$	Verteilung
$W[\varepsilon]$	Gewicht des Niveaus $F_\varepsilon$
<b>Initialisierung</b> einer Population von Suchern $\{x_1 \dots x_s\}$	
<b>FOR</b> $t$ <b>TO</b> $K$	
	<b>FOR ALL</b> $\{x_1 \dots x_s\}$
	$x = \Xi y, \Delta F = F(x) - F(y), z = \text{RANDOM}(0, 1)$
	<b>IF</b> $\Delta > 0$ <b>OR</b> $z < e^{-\beta \Delta F}$ <b>ACCEPT</b> Mutation
	<b>ELSE</b> <b>UNDO</b> Mutation
<b>FOR ALL</b> $\{x_1 \dots x_s\}$	
	<b>IF</b> $F(x_i) \in F_\varepsilon : P[\varepsilon] + 1$
	$W[\varepsilon] = P[\varepsilon] e^{\beta F_\varepsilon}, n_\varepsilon = W[\varepsilon] / \sum_\varepsilon W[\varepsilon]$

Zu unterstreichen ist, daß diese Methode es nicht nur gestattet, die Zustandsdichte approximativ für beliebige Probleme zu bestimmen, sondern gleichzeitig die Optimierung vollzieht. Der Algorithmus wurde als Unterroutine in die im Projekt erstellte Programmbibliothek *ArtEvo* (s. 2.5) zur Optimierung diskreter Probleme implementiert.

Für die Probleme der Netzwerkoptimierung (s. 4.3), der ENGEL-Sequenzen (s. 2.1) und das Problem der Konstruktion minimal korrelierter Sequenzen (LABS, s. 4.2 und 2.3.4) wurden die Zustandsdichten bestimmt. Die Netzwerk- und die LABS-Optimierung sind Minimierungsprobleme, die Fitneß der ENGEL-Sequenzen ist zu maximieren.

Abb. 2.5, 2.6, 2.7 stellen die erhaltenen Dichten semi-logarithmisch dar. Der Exponent  $S(F)$  der Zustandsdichte  $n(F) = e^{-S}$  zeigt im Bereich optima-

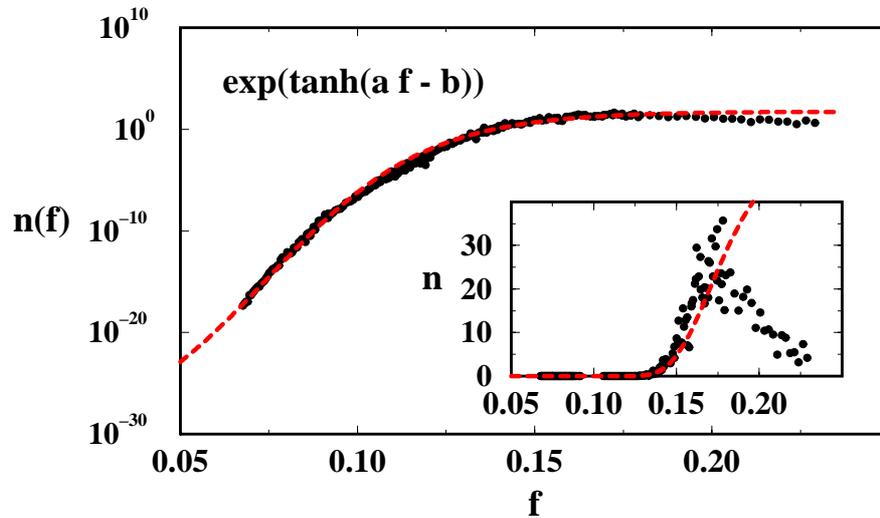


Abbildung 2.5: Zustandsdichte der Netzwerke, 39 Knoten.

ler Fitneßwerte für die Netzwerke einen linearen, für die ENGEL-Sequenzen einen quadratischen und für das LABS-Problem einen höherpolynomialen Verlauf.

Damit erhalten wir folgendes Resultat: Die Zustandsdichte ist ein Klassifikationsmaß der Problemschwierigkeit. Die betrachteten Probleme fallen in drei Klassen steigender Schwierigkeit, die durch den linearen, quadratischen und polynomialen Verlauf des Exponenten der Zustandsdichte beschrieben sind.

### 2.3.2 Das LABS Problem I

(O. Rudzik)

Die im vorigen Abschnitt 2.3.1 beschriebene Methode zur Bestimmung der Zustandsdichte wurde auch auf das LABS (Least Autocorrelated Sequence) Problem angewandt. Da dieses Problem ein in diesem Bericht immer wiederkehrendes Modell ist, soll es hier kurz beschrieben werden.

Die Suche nach Sequenzen mit kleiner Autokorrelationen (LABS), also möglichst chaotischen Sequenzen, ist ein Problem, das in verschiedenen Bereichen der Nachrichtentechnik auftritt. Anwendungen für LABS gibt es z. B. in der Signalverarbeitung, oder in der Radar und Sonartechnik, wo Sequenzen mit stark abfallenden Autokorrelationsfunktionen als Modulationspulse verwen-

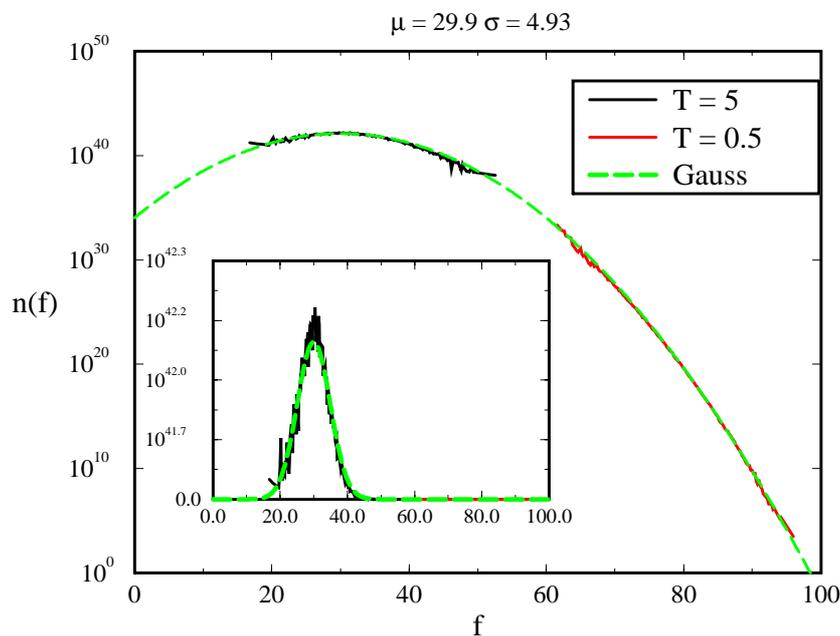


Abbildung 2.6: Zustandsdichte der ENGEL-Sequenzen der Länge  $L = 100$ .

det werden: Um eine hohe Entfernungsauflösung zu erreichen, müssen die Modulationspulse ein möglichst breites Frequenzspektrum umfassen, was genau dann der Fall ist, wenn die Autokorrelationsfunktion stark abfällt. Um auch schwache Ziele erkennen zu können, müssen die Pulse außerdem eine hohe Energie besitzen. Diese Anforderungen werden am besten von einzelnen kurzen Pulsen erfüllt. Für den Fall, daß die Energie für die Einzelimpulse begrenzt ist, kann man als Modulationsimpuls Sequenzen mit stark abfallenden Autokorrelationsimpulsen verwenden. Auf diese Weise ist es möglich, die Energie der Modulationspulse zu erhöhen, ohne die Entfernungsauflösung zu beeinträchtigen.[Ber87]

Für die Zustandsdichte des LABS Problems existieren theoretische Abschätzungen von GOLAY, die auf bestimmten Annahmen (Unabhängigkeit der  $R_k$ ) beruhen. Mit der in Abschnitt 2.3.1 beschriebenen Methode wurde die Zustandsdichte für LABS der Länge 16, 20 und 100 bestimmt, und mit den theoretischen Abschätzungen sowie, soweit möglich, den exakten, durch Abzählen erhaltenen, Daten verglichen.

Die erhaltene Graphen entsprechen qualitativ Abb. 2.7. Es stellt sich eine gute Übereinstimmung der exakten, theoretisch abgeschätzten, und der mit

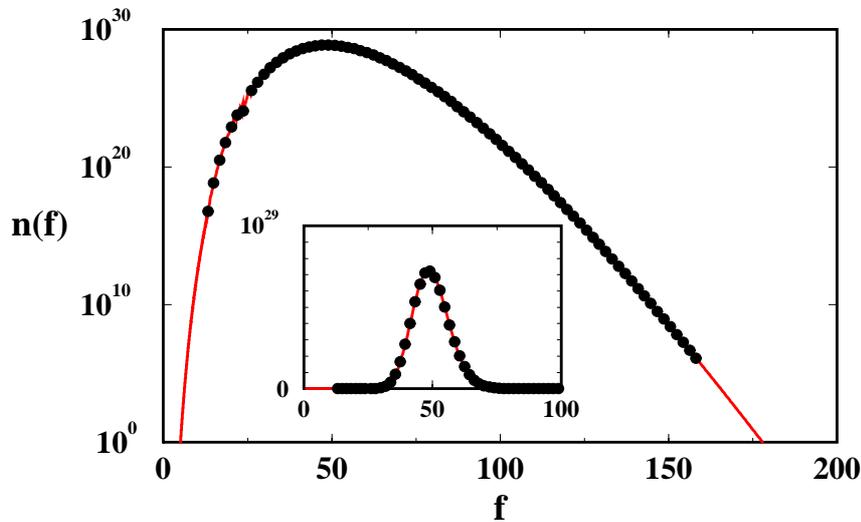


Abbildung 2.7: Zustandsdichte des LABS-Problems, Länge  $L = 100$ .

unserer Methode erhaltenen Werte heraus. Dies ist besonders interessant für große Sequenzlängen, für die keine exakten Werte bekannt sind. Es bedeutet nämlich, daß auch für diese Werte die der theoretischen Abschätzung zugrundeliegenden Annahmen gültig sind.

### 2.3.3 Die Zustandsdichte von Multi-Layer-Perceptronen

Ein Multi-Layer Perceptron (MLP) [HKP91] stellt ein gerichtetes neuronales Netzwerk dar. Seine Neuronen sind in Schichten angeordnet. Daten werden durch eine Eingabeschicht in das Netzwerk eingegeben und durch eine oder mehrere verborgene Schichten geleitet. An der Ausgabeschicht können die Daten abgegriffen werden.

Ein vollvermaschtes MLP mit einem linearen Ausgabeneuron,  $L$  verborgenen (tangens-hyperbolicus-) Neuronen und  $M$  linearen Eingabeneuronen realisiert die Funktion

$$y = \sum_{i=1}^L w_i \tanh\left(\sum_{j=1}^M w_{ij} x_j + w_{i0}\right) + w_0. \quad (2.11)$$

Die Variablen  $\vec{w}$  repräsentieren die Gewichte des Netzes. Die Schwellwerte  $w_0$  der Neuronen können als externe Neuronen mit festen Eingaben ( $= -1$ ) implementiert werden. Das MLP stellt somit einen nichtlinearen funktionalen Zusammenhang zwischen Eingavektoren  $\vec{x} \in \mathbb{R}^M$  und Ausgaben  $y \in \mathbb{R}$  her.

Es kann daher zur Approximation funktionaler Zusammenhänge eingesetzt werden.

MLPs haben als einfache Modelle für komplexe lernende Systeme als auch als effiziente Werkzeuge zur Approximation Eingang in die Physik und in die Technik gefunden.

Der Einsatz von MLPs gliedert sich in zwei Phasen, die Lern- und die Generalisierungsphase. In der ersten Phase werden die Gewichte  $\vec{w}$  des Netzes mit Hilfe eines Datensatzes  $D_L$  gelernt. Der Datensatz setzt sich zusammen aus  $n$  Beispielpaaren für den Zusammenhang zwischen Eingabevektoren  $\vec{x}$  und Zielwerten  $t$

$$D_L := \{(\vec{x}_1, t_1), \dots, (\vec{x}_n, t_n)\} . \quad (2.12)$$

Das Ziel des Lernens ist die korrekte Abbildung jedes Paares aus Gl. (2) entsprechend Gl. (1). Der Lernvorgang entspricht der Minimierung des mittleren quadratischen Fehlers im Raum der Gewichte

$$\vec{w} = \arg \min_{\vec{w}} \frac{1}{n} \sum_{i=1}^n (y_i - t_i)^2 . \quad (2.13)$$

Die Adaptation der Gewichte kann über einen einfachen Gradientenabstieg erfolgen. Nach Adaptation der Gewichte stellt das Netz eine nichtlineare Approximationsfunktion  $y = f(\vec{x}, \vec{w})$  für den wahren Zusammenhang  $y_0 = f(\vec{x}_0)$  dar.

Nach dem Lernvorgang ist Wissen über die Struktur der Daten in den Gewichten des Netzes repräsentiert. Aus diesem Grund kann das Netzwerk Aussagen  $v$  zu nicht gelernten Eingaben treffen. Dieser Vorgang wird Verallgemeinerung genannt. Wird ein Teil der Daten dem Lernen vorenthalten, können empirische Aussagen zur Verallgemeinerungsfähigkeit gemacht werden. Nachdem das Netz gelernt wurde, werden diese Daten zum testen der Verallgemeinerungsfähigkeit genutzt. Diese Aufspaltung der gesamten Datenmenge führt zu den Teilmengen  $D_L$  (Lernmenge) und  $D_V$  (Verllgemeinerungsmenge). Der Verallgemeinerungsfehler  $E_V$  auf der Menge  $D_V$  wird berechnet über

$$E_V = \frac{1}{m} \sum_{j=1}^m (v_j - t_j)^2 . \quad (2.14)$$

Kann nur ein kleines und veraushtes Sample der Funktion  $y_0$  zum Lernen genutzt werden, so zeigt sich für die Güte des Verallgemeinerungsfehlers eine

starke Abhängigkeit vom Verknüpfungsmuster der Gewichte (Netzarchitektur), von der Grösse der Lernmenge, sowie von der Anzahl der Lernschritte. Dieses Phänomen wird Overlearning genannt. Wird der Lernvorgang als parametrische Regression aufgefasst, entspricht das Overlearning einer Überanpassung der Gewichte (Freiheitsgrade des Netzes) an die Lernmenge. Das Overlearning kann durch eine Optimierung der Netzarchitektur vermieden werden. Ziel dieser Optimierung ist die Ausdünnung der Gewichte des Netzes bis zu einem Grad, bei dem einerseits genügend Gewichte für die Approximation zur Verfügung stehen und andererseits die Überanpassung vermieden wird. Als Methoden für die Architekturoptimierung haben sich das Pruning [SHJ93] oder die Optimierung mittels evolutionärer Algorithmen [HSG90] bewährt. Gemeinsames Merkmal aller Methoden ist die Suche im Raum der Architekturen mit dem Ziel der Minimierung des Verallgemeinerungsfehlers.

Ziel des Projektes war eine Analyse der Zustandsdichte von MLP-Architekturen [REA96]. Sie erlaubt Einblick in den Schwierigkeitsgrad der Architekturoptimierung. Hierzu wurde obiges Lernverfahren implementiert und an Routinen zur Analyse der Zustandsdichte angebunden (C-Code).

Die Architektur der Netze wird in Form von binären Vektoren  $\vec{x} \in \{0, 1\}^K$  kodiert. Die Zahl  $K$  entspricht der Anzahl von Gewichten. Jeder Komponente von  $\vec{x}$  entspricht eindeutig ein Gewicht des Netzes. Eine 1 in  $\vec{x}$  entspricht einem existenten Gewicht, eine 0 einem nicht existenten Gewicht. Somit entspricht jedes  $\vec{x}$  einer der  $2^K$  möglichen Architekturen des Netzes.

Zur Analyse der Zustandsdichte wird ein entsprechendes  $\vec{x}$  vorgegeben, das Netz eine feste Schrittzahl gelernt und anschliessend der Lern- bzw. der Verallgemeinerungsfehler berechnet. Der sich anschliessende Zustandsübergang  $\vec{x} \rightarrow \vec{x}'$  erfolgt in Abhängigkeit vom Verallgemeinerungsfehler.

Als Datenmenge wurde die Sunspot-Zeitreihe gewählt [SHJ93]. Die Reihe beschreibt die jährliche mittlere Zahl von Sonnenflecken für den Zeitraum von 1700–1979. Als Lernmenge  $D_L$  wird üblicherweise der Zeitraum von 1700–1920, als Verallgemeinerungsmengen die Zeiträume  $D_{V,1}$  von 1921–1955 und  $D_{V,2}$  von 1956–1979 verwendet.

Ziel des Lernens ist die korrekte Vorhersage der Sonnenfleckenaktivität. Aus den Werten der  $M$  letzten Zeitpunkte der Vergangenheit (entspricht den  $M$  Eingabeneuronen) soll der aktuelle Wert (entspricht einem Ausgabeneuronen) auf der Lernmenge gelernt werden. D.h. es wird ein Fenster der Grösse  $M + 1$  über die Zeitreihe geschoben, wobei das Netz eine Abbildung der  $M$  letzten Datenpunkte auf den Aktuellen darstellt. Das gelernte Netz soll den aktuellen Wert auf der Verallgemeinerungsmenge korrekt vorhersagen können. Diese Vorgehensweise – auf den Sunspot Daten – hat sich auf dem

Gebiet der neuronalen Netze als Standard zur Beurteilung von Verfahren zur Steigerung der Verallgemeinerungsfähigkeit durchgesetzt. Im folgenden wird eine Untermenge von  $D_L$  zum Lernen und  $D_{V,2}$  zur Auswertung des Verallgemeinerungsfehlers verwendet.

Einführend sei gesagt, dass für die Berechnung jeder einzelnen Architektur ein voller Lernzyklus durchlaufen werden muss. Die Messung der Zustandsdichte erfordert somit grosse 'Rechenzeiten'. Die folgenden Analysen wurden daher für relativ kleine Netze durchgeführt. Weiterhin wird die Zustandsdichte nur mit geringer Auflösung dargestellt.

Zunächst werden Resultate für Netzwerke mit 3 Eingabeneuronen, 3 verborgenen Neuronen und 1 Ausgabeneuron vorgestellt. Ein vollvermaschtes Netz besteht aus 16 Gewichten, die Zahl der möglichen Architekturen beträgt somit  $2^{16} = 65536$ . Die Startwerte der Gewichte (d.h. Startwerte des Lernvorgangs) wurden zufällig aber fest (konstante seed für den Zufallszahlengenerator) gewählt. Somit wird der Raum der Architekturen für den gleichen Satz von Startgewichten untersucht. Die Lernmenge beinhaltet den Zeitraum von 1700–1760. Die Zahl der Lernschritte beträgt 5000. Bei dieser Zahl von Lernschritten kann Overlearning beobachtet werden.

Abbildung 2.8 zeigt die Zustandsdichte. Die Abbildung wird in Richtung steigender Verallgemeinerungsfehler (links (gute Netze)  $\rightarrow$  rechts (schlechte Netze)) diskutiert. Die Zustandsdichte steigt zunächst an und fällt zu schlechteren Netzen hin ab. Sehr gut verallgemeinernde Netze sind deutlich schwerer zu finden als nur gut verallgemeinernde. Von Interesse ist der strukturierte Schwanz zu schlechten Netzen hin. Die Struktur ist interpretierbar. Bei den betrachteten Netzen handelt es sich um gerichtete Netzwerke. Eine ausreichend gute Abbildung (guter Informationsfluss) ist nur möglich für Netze mit hinreichender Dichte (bzgl. der Gewichte). Betrachten wir den Extremfall des Fehlens aller Gewichte zwischen verborgener und Ausgabeschicht (das Netz liefert hier  $E_V = 0.82$ ). In diesem Fall können viele Architekturen zum gleichen Wert von  $E_V$  realisiert werden, durch beliebige Verknüpfungen von Gewichte zwischen Eingabe- und verborgener Schicht. Insbesondere bei Hinzunahme der Schwellwerte (externe Neuronen) können in dieser Art sehr viele Zustände zu hohen Werten von  $E_V$  realisiert werden. Von besonderem Interesse für die Optimierung ist der Umstand, dass die Dichte zu hohen Werten von  $E_V$  näherungsweise der Dichte für sehr niedrige Werte von  $E_V$  entspricht.

Zusätzlich werden Resultate für Netzwerke mit 8 Eingabeneuronen, 4 verborgenen Neuronen und 1 Ausgabeneuron vorgestellt. Ein vollvermaschtes Netz besteht aus 41 Gewichten, die Zahl der möglichen Architekturen beträgt somit  $2^{41}$ . Die Lernmenge beinhaltet den Zeitraum von 1700–1820. Es

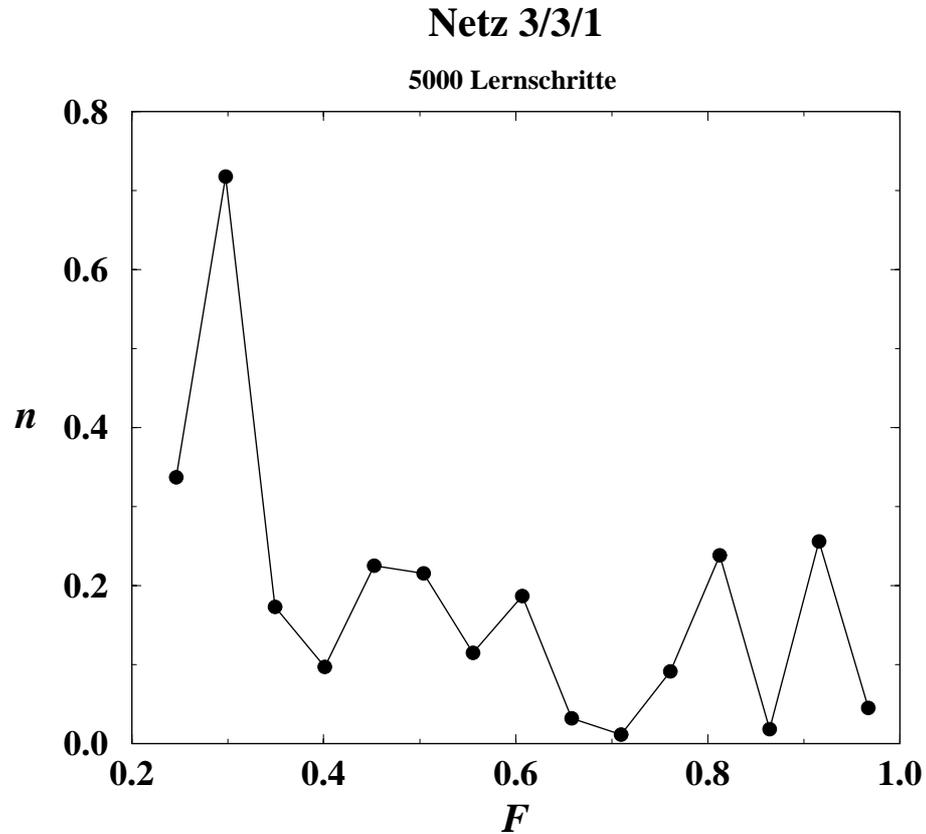


Abbildung 2.8:

werden zwei Zahlen von Lernschritten betrachtet, 500 und 5000. Bei der Zahl von 500 Lernschritten kann noch kein Overlearning beobachtet werden. Bei der Zahl von 5000 Lernschritten kann bereits Overlearning beobachtet werden.

Abbildung 2.9 zeigt die Zustandsdichte für den Verallgemeinerungsfehler bei einer Zahl von 5000 Lernschritten. Die Interpretation der Struktur der Kurve kann analog zu obiger Interpretation geführt werden.

Neue Resultate sind bei Betrachtung der Abbildung 2.10 zu erkennen. Die Abbildung zeigt die Zustandsdichte für den Verallgemeinerungsfehler bei einer Zahl von 500 Lernschritten. Insbesondere ist der fehlende Abfall für sehr gute Werte zu erkennen. Die geringe Zahl von Lernschritten erlaubt grossen Netzen keine Überanpassung und kleinen Netzen kein ausreichendes Lernen. Insgesamt können daher keine Netze mit sehr niedrigem Verallgemeinerungsfehler (vergleiche dazu Abbildung 2.10) und somit kein Abfall gefunden werden.

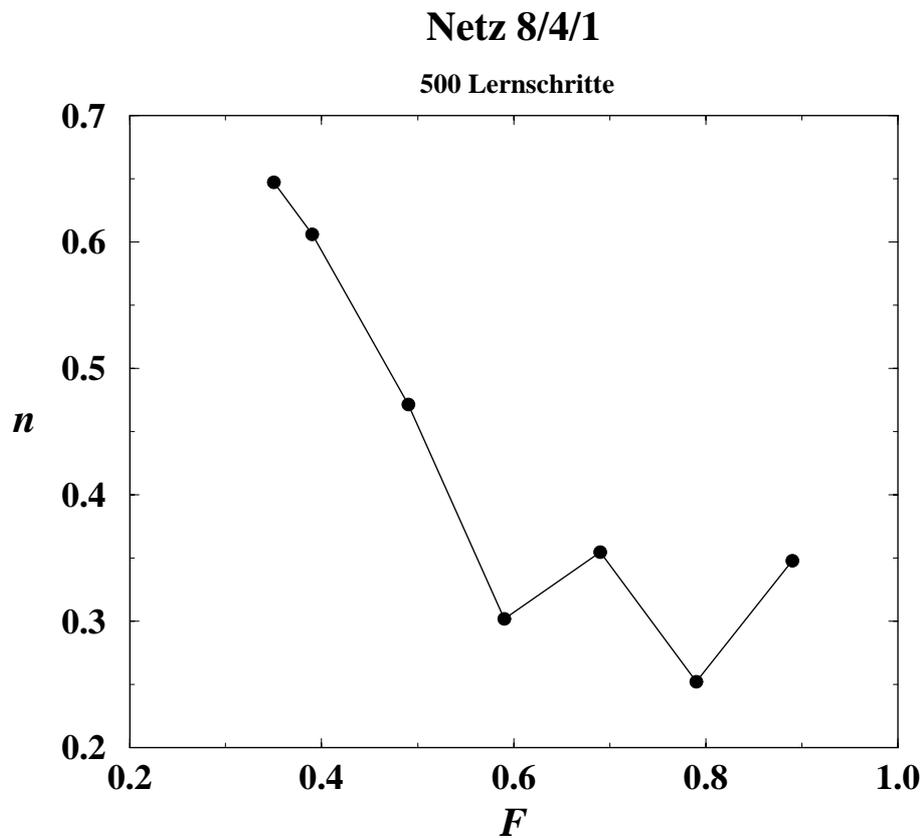


Abbildung 2.9:

### 2.3.4 Eine Methode zum Vergleich von Mutationsoperatoren

(C. Kappler, M. Zamparelli)

## 2.4 Klassifikation von Optimierungsstrategien

(T. Afelmeyer)

In diesem Abschnitt soll eine Klassifikation der Fitnessfunktionen in Bezug auf einen Algorithmus durchgeführt werden. Dabei steht die Frage nach den

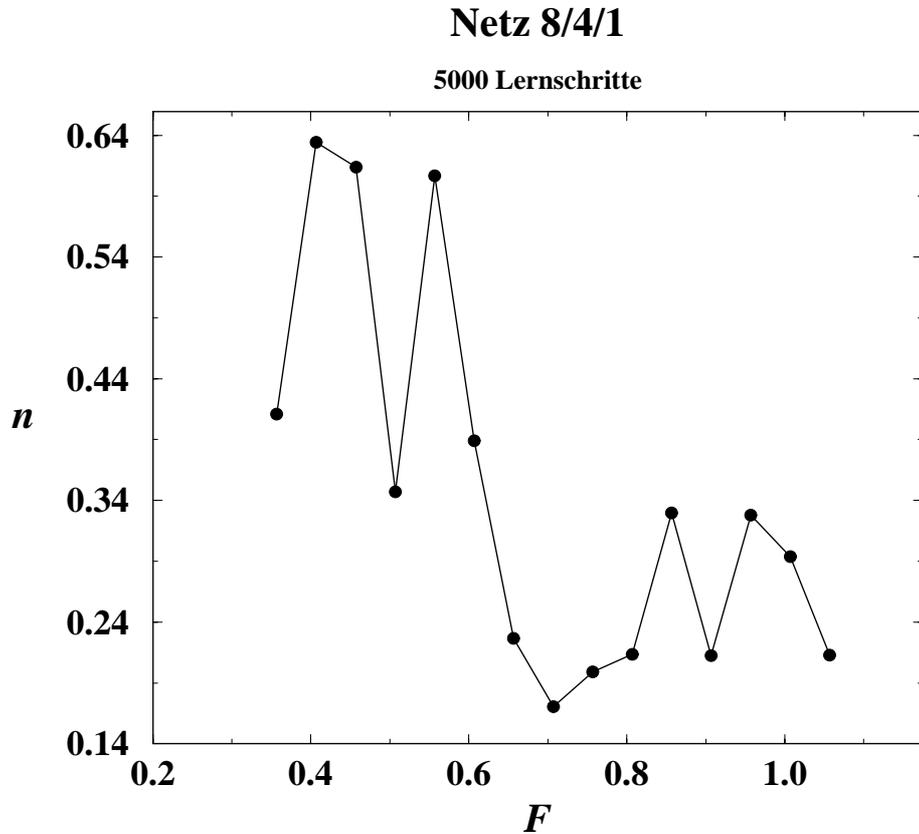


Abbildung 2.10:

schwierigen Problemen im Mittelpunkt, d.h. in welchen Fällen hat der Algorithmus Schwierigkeiten ein Optimum zu finden und wann nicht. Eine anschauliche Vorstellung von solchen Problemen besitzt natürlich jeder. So ist sicherlich ein unimodales Problem schwieriger als ein Doppeltopf mit zwei lokalen Minima. Erstaunlicherweise wird sich herausstellen, daß die Anzahl der schwierigen Fälle endlich ist und die sogenannten Thomschen Katastrophen umfaßt.

Dazu betrachten wir zuerst die Darwin, Boltzmann und Gemischte Strategie in einem kontinuierlichen Suchraum  $B^d$  der Dimension  $d$ . Eine Zusammenstellung der entsprechenden Untersuchungen zu diesen 3 Strategien kann man im Kapitel 4 (Entwicklung optimaler Algorithmen) finden. In einem (im Projekt entstandenen) Artikel [AE97] wurden sowohl die thermodynamische Strategie (Boltzmann Strategie) als auch die biologische Strategie (Darwin Strategie) auf eine einheitliche Grundlage gestellt. Beide Strategien konnten

nach entsprechenden Transformationen auf die Gleichung

$$\frac{d}{dt}P(\vec{x}, t) = -HP(\vec{x}, t), \quad (2.15)$$

wobei  $H$  ein positiv definit, selbstadjungierter Operator der Form:

$$H = \sum_{i,j} a_{ij} \frac{\partial^2}{\partial x_i \partial x_j} + F(\vec{x}) \quad (2.16)$$

und  $F(\vec{x})$  die Fitnessfunktion  $F : B \rightarrow \mathbb{R}$  über den Raum  $B$  ist, zurückgeführt werden. Als formale Lösung ergibt sich

$$P(\vec{x}, t) = \exp(-tH)P(\vec{x}, 0) \quad (2.17)$$

mit  $\exp(-tH)$  als sogenannten Wärmeleitungskern (Heat-Kernel). Andererseits kann man die Lösung von (2.15) über die Eigenwerte des Operators  $H$  (siehe Abschnitt 4.1) darstellen, d.h.

$$H\phi_i(\vec{x}) = \lambda_i\phi_i(\vec{x}) \quad (2.18)$$

woraus sich die Lösung

$$P(\vec{x}, t) = \sum_i \phi_i(\vec{x}) \exp(-\lambda_i t) \quad (2.19)$$

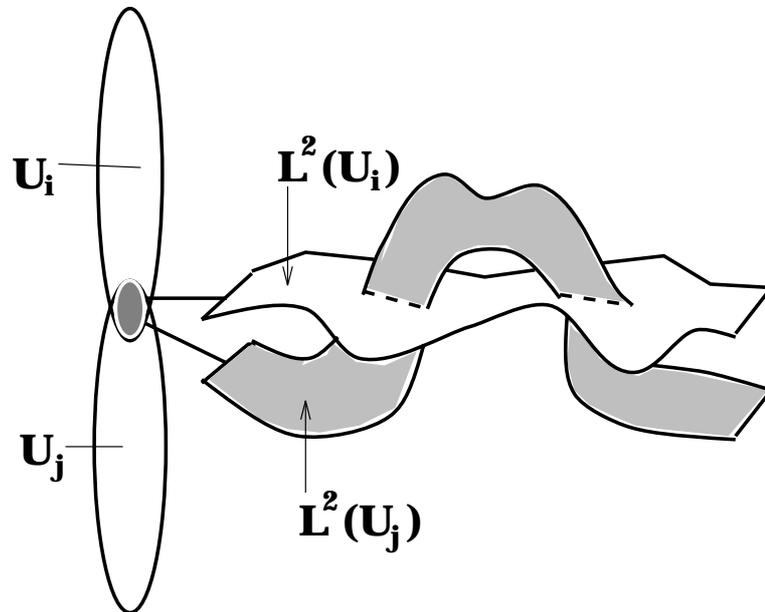
ergibt. Durch die Methoden der Funktionalanalysis kann gezeigt werden, daß die Lösung stetig von der Fitness  $F$  abhängt. Diese Aussage ist aber nicht für alle Funktionen  $F$  richtig. Deshalb werden uns die qualitative Änderungen der Lösungen, die auch als *Bifurkationen* bezeichnet werden, interessieren.

Entsprechend der Sturm-Liouvilleschen Theorie bilden die Eigenfunktionen  $\phi_i(\vec{x})$  einen Hilbertraum, der gleichzeitig der Lösungsraum der Gleichung (2.15) ist. Leider hat dieser Lösungsraum einen entscheidenden Nachteil: Um ihn zu bestimmen muß man das vollständige Eigenwertproblem (2.18) lösen. Für die meisten Probleme ist das aber nicht möglich und man muß sich eine praktikablere Beschreibung für diesen Lösungsraum einfallen lassen. Wir bemerken dazu, daß man in der Evolution meist Fitnessfunktionen (Potentiale)  $F(\vec{x})$  betrachtet, die sich in eine Potenzreihe

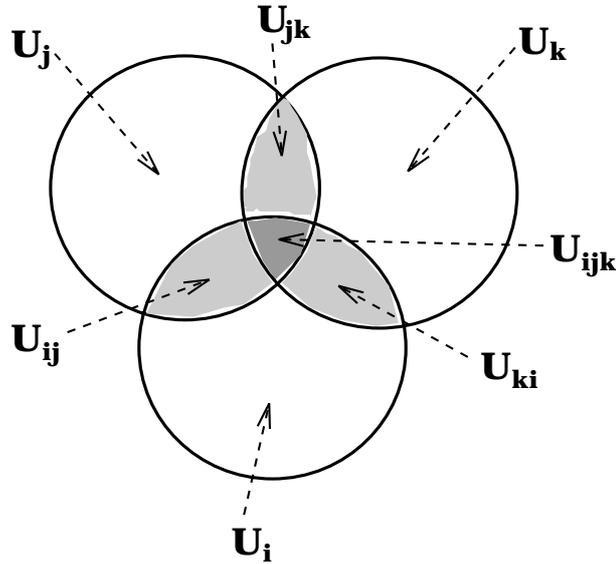
$$F(\vec{x}) = F(\vec{x}_0) + \nabla F(\vec{x}_0)(\vec{x} - \vec{x}_0) + \frac{1}{2} \Delta F(\vec{x}_0)(\vec{x} - \vec{x}_0)^2 + \dots \quad (2.20)$$

entwickeln lassen. Sei weiterhin der Phänotypiraum  $B \subset \mathbb{R}^n$  eine  $n$ -dimensionale kompakte Untermenge des  $\mathbb{R}^n$  mit  $\partial B$  homöomorph zu  $S^{n-1}$ , dann finden wir eine Zerlegung von  $B$  in offene Untermengen  $U_i$ , so daß  $F(\vec{x})$  ein Polynom

vom Grad  $p$  ist (natürlich bis auf einen kleinen Fehler). Die Untermengen  $U_i$  und  $U_j$ , die benachbart sind, müssen sich überlappen, d.h. wir erhalten eine offene Überdeckung von  $B$  mit Karten  $U_i$ . Es ist klar, daß diese Zerlegung vom Grad des Polynoms abhängt. Für die Konstruktion des Lösungsraumes wählen wir  $p = 2$ , da wir dann eine explizite Lösung für das Eigenwertproblem (2.18) bzw. für die gesamte Gleichung (2.15) besitzen. Dadurch bekommen wir über jeder Karte  $U_i$  einen Hilbertraum  $L^2(U_i)$ . Aufgrund der Überlappung zwischen zwei Karten sind diese Hilberträume nicht unabhängig voneinander. Man kann nun zeigen, daß der Lösungsraum eine Blätterung über  $B$  mit den Hilberträumen  $L^2(U_i)$  als Blättern ist. Bild 2.11 zeigt eine Beispielsituation. Wir wollen nun die Grundidee für die Klassifi-

Abbildung 2.11: Blätterung über  $B$ 

kation vorstellen. Sei  $\mathcal{F}$  die oben beschriebene Blätterung und  $\{U_i\}_{i \in I}$  die offene Überdeckung von  $B$ . Wir betrachten 3 Karten  $U_i$ ,  $U_j$  und  $U_k$  mit  $U_i \cap U_j \cap U_k \neq \emptyset$  (siehe Bild 2.12). Auf jeder Karte ist die Lösung des Problems bekannt, aber die gegenseitige Beziehung zwischen den Lösungen ist noch nicht klar. Dazu wählt man eine Transformation von einer Karte zur anderen, die auch als Übergangsabbildung bezeichnet wird. Bezeichne  $U_{ij}$  den Überlapp  $U_i \cap U_j$  zwischen  $U_i$  und  $U_j$ . Die Verklebung zweier Teillösungen auf den Karten erfolgt durch die Festlegung einer Funktion  $g_{ij} : U_{ij} \rightarrow GL(n, \mathbb{R})$  (analog für die anderen Kartenübergänge) auf die Gruppe der linearen Transformationen. Aufgrund der Gruppenstruktur der

Abbildung 2.12: Darstellung von 3 Karten in  $B$ 

$GL(n, \mathbb{R})$  muß  $g_{ij} = g_{ik}g_{kj}$  gelten. Wir bemerken, daß die Wahl der Gruppe keine Einschränkung darstellt, da wir immer noch die Freiheit haben, auf den Karten noch beliebige differenzierbare Kartentransformationen durchzuführen. Wir sprechen von einer trivialen Transformation  $g_{ij}$ , wenn zwei Funktionen  $f_i : U_i \rightarrow GL(n, \mathbb{R})$  und  $f_j : U_j \rightarrow GL(n, \mathbb{R})$  existieren, so daß  $g_{ij} = f_i^{-1}f_j$  gilt. Dann sind nämlich die beiden Lösungen auf den Karten voneinander unabhängig. Die Menge der nichttrivialen Transformationen  $g_{ij}$  wird mit  $\check{H}^1(\{U_i\}_{i \in I}, GL(n, \mathbb{R}))$  bezeichnet und heißt 1. Čechische Kohomologie mit Werten in  $GL(n, \mathbb{R})$ . Mittels komplizierter algebraischer Methoden (K-Theorie [Ati67], rationale Homotopietheorie [GM81]) kann diese Menge untersucht werden (siehe z.B. [Fre88] für die Verbindung zu Fredholm-Operatoren). Entscheidend ist nun die Tatsache, daß wir lokal den Operator  $H$  als quadratische Form darstellen können, und die Lösungen des Operator eng mit dieser Darstellung verknüpft sind. Die stabile Klassifikation dieser lokal definierten, quadratischen Formen erzeugt eine Einteilung des unterliegenden Raumes in 4-dimensionale Teilräume. Da jeder Operator durch die Wahl der Fitnessfunktion gegeben ist, überträgt sich diese Einteilung auf den Raum der Fitnessfunktionen. Das bedeutet aber, daß wir eine Aufspaltung der Fitness in der Form:

$$F(x_1, \dots, x_8) \longrightarrow \tilde{F}(x_1, \dots, x_4) + \tilde{F}(x_5, \dots, x_8)$$

und damit eine Zerlegung des Eigenwertproblems (2.18) in zwei 4-dimensionale

Probleme erhalten. Um die Klassifikation abzuschließen, muß nun noch die Abhängigkeit des Eigenwertproblems von einer 4-dimensionalen Fitness untersucht werden. Dazu stellen wir erst einmal fest, daß eine zweimal-differenzierbare Transformation des Operators  $H$  zu einer stetigen Änderung des Spektrums führt. Der Laplace-Operator ändert dabei das Verhalten nicht so sehr wie die Fitness  $F(\vec{x})$ . Wie ist nun aber der Raum der Fitnessfunktionen, d.h. der Raum der Funktionen  $B \rightarrow \mathbb{R}$ , aufgebaut? Die Antwort läßt sich mit der Singularitätstheorie [GG73] geben: Der Raum wird durch 6 Thomsche Katastrophen aufgespannt (Thom-Boardman-Stratifizierung), d.h. diese Katastrophen liegen dicht im Raum der Funktionen  $\mathbb{R}^4 \rightarrow \mathbb{R}$ . Wir erhalten diese 6 Katastrophen als die stabilen Äquivalenzklassen von Funktionen  $\mathbb{R}^4 \rightarrow \mathbb{R}$ , die sich als  $(x_1, x_2, x_3, x_4) \rightarrow (x_1, x_2, x_3, F(x_1, x_2, x_3, x_4))$  darstellen lassen, d.h. diese Katastrophen liegen dicht im Raum der Funktionen  $\mathbb{R}^4 \rightarrow \mathbb{R}$ . Durch die obige Aussage über die Aufspaltung der Richtungen erhält man überhaupt erst die endliche Anzahl von Fällen.

Um diese Aussage nicht zu trocken zu gestalten, wollen wir noch einige physikalische Argumente anbringen, die dieses Ergebnis plausibel erscheinen lassen. Dazu betrachtet man die Gleichung (2.15) mit dem Operator  $H = -\Delta + F$  und konstruiert als Lösung das Pfadintegral nach Feynman [Fey48]:

$$P(\vec{x}, t) = \int D\vec{x} \exp \left( - \int_B (\dot{x}^2 + F) d^n x \right) . \quad (2.21)$$

Eine Sattelpunktapproximation zeigt, daß der größte Beitrag zu diesem Integral durch den Pfad

$$\frac{d^2}{dt^2} \vec{x} = -\nabla F(\vec{x}) \quad (2.22)$$

erbracht wird. Das ist aber genau ein Gradientensystem, für welches THOM seine Theorie entwickelt hat.

Doch nun zurück zu den Thomschen Katastrophen. Im Gegensatz zu THOM betrachten wir keine Entfaltungen, d.h. wir teilen nicht die Variablen in Parameter und Ort auf. Bis auf die linearen Terme betrachten wir alle Parameter auch als Ortsvariablen. Nimmt man z.B. die Entfaltung  $x^3 - ux$  der Katastrophe  $x^3$ , so bleibt  $u$  ein Parameter während im Fall des Doppelpfades  $x^4 - yx^2$  die Variablen  $x, y$  einen Ort bezeichnen. Dadurch ergeben sich Katastrophen in 1,2,3 und 4 Dimensionen. Andererseits kann die zweidimensionale Katastrophe  $x^4 - yx$  auch in 4 Dimensionen auftreten. Dazu müssen wir einfach die anderen Richtungen durch quadratische Terme "auffüllen" (stabile Äquivalenz in der Singularitätstheorie). Die folgende Tabelle gibt über die einzelnen Fälle Aufschluß.

Name der Katastrophe	Formel	Dimension	
		Raum	Parameter
Falte	$x_1^3 + u_1x_1$	1	1
Spitze	$x_1^4 + x_2x_1^2 + u_1x_1$	2	1
Schwalbenschwanz	$x_1^5 + x_2x_1^3 + x_3x_1^2 + u_1x_1$	3	1
Schmetterling	$x_1^6 + x_2x_1^4 + x_3x_1^3 + x_4x_1^2 + u_1x_1$	4	1
hyperbolischer Nabel	$x_1^3 + x_2^3 + x_3x_1x_2$ $-u_1x_1 - u_2x_2$	3	2
elliptischer Nabel	$x_1^3 - x_1x_2^2 + x_3(x_1^2 + x_2^2)$ $-u_1x_1 - u_2x_2$	3	2

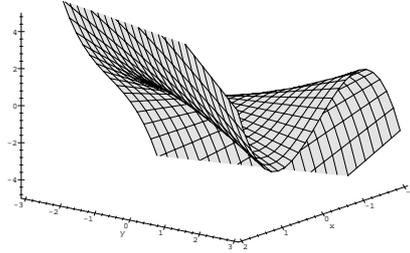
Damit können wir das Ergebnis dieses Arbeitspunktes zusammenfassen:

*Zwei Fitnesslandschaften sind bzgl. der Strategien vom Typ (2.16) gleich, falls sie aus genau denselben Katastrophen bestehen.*

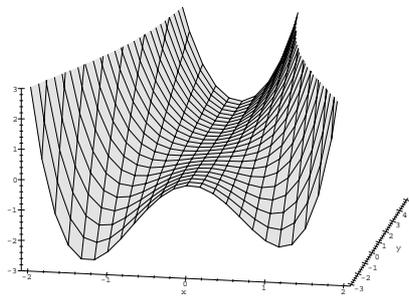
Evolutionäre Algorithmen lassen sich durch die Katastrophentheorie bzgl. ihres Verhaltens klassifizieren.

Um eine Vorstellung von der Form der Fitnessfunktionen zu bekommen, wollen wir noch kurz auf diese Katastrophen eingehen. Im Bild 2.13 sind die einzelnen Fälle dargestellt, wobei die Darstellung der höher-dimensionalen Katastrophen (ab Dimension 3) durch Projektion auf 2 Dimensionen erhalten wurde. Die ersten beiden Fälle (Falte und Spitze) repräsentieren das einfachste Problem: der Algorithmus muß einen "Wall" überspringen. Diese Art von Problemen ist in vielen Fällen der am häufigsten auftretende Fall (z.B. Rastriginfunktion). Desweiteren können natürlich Probleme existieren, die den lokal arbeitenden Algorithmus "täuschen". GOLDBERG [Gol89b] betrachtet auch solche "misleading problems" im Falle der genetischen Algorithmen. Durch diese Klassifikation konnten seine Arbeiten auf diesem Gebiet untermauert werden. So folgt der Algorithmus in den meisten Fällen der Richtung, die die größte Verbesserung verspricht. Das kann aber genau die falsche Richtung sein, wenn man das globale Optimum finden will. In diese zweite Klasse von Problemen fallen die anderen Fälle. Sie repräsentieren die einfachsten Möglichkeiten für solche "irreführenden" Probleme und zeigen vor allem, wie die Umgebung des Punktes beschaffen ist, in der der Algorithmus die Entscheidung treffen muß, welche Richtung er einschlagen will.

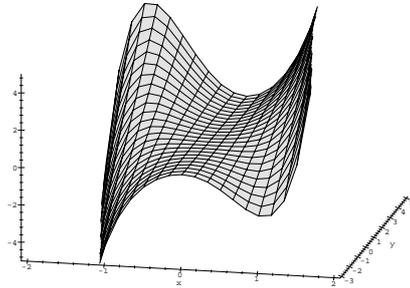
Die oben beschriebene Klassifikation evolutionärer Algorithmen basiert auf der genauen Analyse der Dynamik dieser Algorithmen. Bei praktischen Problemen hat man es manchmal auch mit kontinuierlichen Suchräumen zu tun. An dieser Stelle sollte man die Klassifikation verwenden, um zu einer besseren Steuerung der Parameter des Algorithmus und damit zu einem



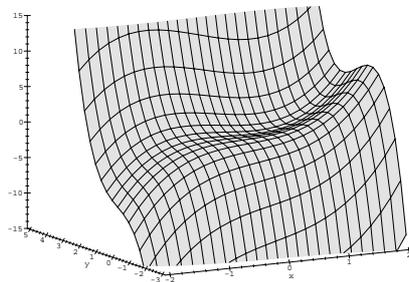
**Falte**



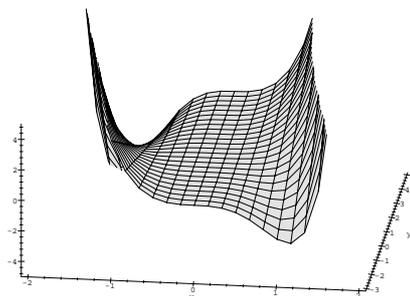
**Spitze**



**Schwabenschwanz**



**hyperbolischer Nabel**



**Schmetterling**

besseren Algorithmus zu gelangen. Das kann auf zwei möglichen Wegen geschehen. Als erstes zeigt die Klassifikation, daß die Berechnung der Geschwindigkeiten für den Fall der Parabel  $x^2$  am wichtigsten ist, da die Katastrophen aus dieser Funktion aufgebaut sind. Damit können die Parameter des Algorithmus mit Hilfe der berechneten Geschwindigkeiten der Parabel (siehe 4.1) gesteuert werden. Der zweite Weg besteht in der Entwicklung des Wärmeleitungskernes (2.17) nach dem Zeitparameter  $t$ . Der erste Term dieser Entwicklung ( $t = 0$ ) entspricht der Zustandsdichte (siehe 2.4). Die weiteren Terme stehen im Zusammenhang mit den lokalen Eigenschaften der Fitnessfunktion. So ist der lineare Term (linear in  $t$ ) einfach die zweite Ableitung der Fitnessfunktion und damit identisch mit der lokalen Krümmung der Landschaft. Statt den Wärmeleitungskern zu entwickeln, kann man auch die Lösung (2.19) betrachten. Nun wird das Verhalten der Lösung durch die Form des Spektrums bestimmt, d.h. welchen funktionalen Zusammenhang gibt es zwischen einem freien Parameter und den Eigenwerten. Für die Parabel ist diese Funktion linear. Ein steilerer Anstieg der Funktion bedeutet i.a., daß die Strategie schneller ist als im Falle der Parabel. Die Eigenwerte können während der Simulation näherungsweise bestimmt werden und lassen so erkennen, in welchem Potential vom oben genannten Typ man sich befindet. Die Untersuchungen zu diesem Thema sind noch nicht abgeschlossen, werden aber teilweise in dem geplanten Artikel [Ass96] veröffentlicht. Der hauptsächliche Nutzen der Klassifikation besteht darin, daß man nun entscheiden kann, wann Algorithmen gut arbeiten. Dies ist dann der Fall, wenn sie alle oben genannten Katastrophen problemlos meistern. Für unsere Algorithmen ist das erfüllt und mit der neu entwickelten Gemischten Strategie (siehe 4.1) steht auch ein Algorithmus zur Verfügung, der schneller und robuster als die anderen Strategien ist.

### **Ergebnis:**

Für die wichtigsten Strategien wurde eine Klassifikation nach ihrem Verhalten in Abhängigkeit von der Fitness erhalten. Damit können gleichzeitig auch die für eine Strategie schwierigen Fälle bestimmt werden. Für die Anzahl solcher Fälle konnte gezeigt werden, daß sie endlich ist und nur 6 Typen umfaßt, die als Thomsche Katastrophen bekannt sind. Weiterhin konnte gezeigt werden, daß es Maße geben muß, die in praktischen Problemen benutzt werden können, um den Schwierigkeitsgrad eines Problems abzuschätzen und Strategien zu steuern. Ein solches Maß ist die Zustandsdichte.

## 2.5 Erstellte Software

(H. Rosé)

### 2.5.1 Seq

NAME: seq [-r]

OPTIONEN:

-r : Runmode, interaktive Eingabe deaktiviert.

BESCHREIBUNG:

Das Programm implementiert Routinen zur Bestimmung der in 2.2, 2.3.1 und [AER96] eingeführten Größen zur Klassifikation (Schema, Schemavektoren, Phänotypen, Zustandsdichte) für das Problem der *Engel*-Sequenzen (s. 2.1). Die Parametereingabe erfolgt interaktiv oder über ein Parameterfile. Folgende Funktionen wurden realisiert:

- *Darwinstrategie*: Simulation der Darwinstrategie und Berechnung der Transinformation der Sequenzen.
- *Fitneßniveaus*: Darwinstrategie mit Speicherung aller gefundenen Sequenzen in einem definierten Fitneßintervall.
- *Fitneß Scan*: Vollständige Enumeration der Fitneßlandschaft (nur sinnvoll für kurze Sequenzen).
- *Schema Scan*: Vollständige Enumeration der Fitneßlandschaft und Speicherung aller Schemata, Schemavektoren, Phänotypen und Besetzungszahlen (Zustandsdichte) für kurze Sequenzen.
- *Zustandsdichte*: Bestimmung der Zustandsdichte langer Sequenzen mittels der Boltzmannstrategie.
- *Hamilton Scan*: Vollständige Enumeration der Valuationslandschaft auf einem Hamiltonweg (aufeinander folgende Sequenzen besitzen im-

mer den Hammingabstand eins) und Bestimmung des Stetigkeitsgrades (nur sinnvoll für kurze Sequenzen).

**PARAMETER:** Parameterfile: seq1.4.para

%00 Populationsgröße  
%01 Sequenzlänge  
%02 Periode der Fitneßfunktion  
%03 Generationen  
%04  $b$  Periodenfitnessfaktor  
%05 Mutationsrate  
%06 Selektionsrate  
%07 Punktmutationsrate (nicht benutzt)  
%09 Alphabet (abcd)  
%10 Temperatur  
%11 Obere Fitneßschranke  
%20 Speicherperiode  
%22 Runmodus  
%23 Sortierflag (0,1)  
%24 Minimalwert des Fitneßniveaus  
%25 Maximalwert des Fitneßniveaus  
%26 Speicherintervall der Transinformation  
%27 Anzahl ausgedruckter Sequenzen  
%28 Ausgabeperiode  
%29 Anzahl der Runs  
%30 Datenmode (0 : Datenfile nicht einlesen)  
%31 Seed des Randomgenerators (0 : zufällig)  
%32 Datenfile der Startsequenz

**AUTOR:** Helge Rosé (rose@summa.physik.hu-berlin.de)



# Kapitel 3

## Optimale Algorithmen und adaptive Strategien

### 3.1 Die Boltzmann Strategie

(T. Aßelmeyer)

Als einfachste Variante einer Evolutionären Strategie wollen wir zunächst die Boltzmann Strategie betrachten. Sie wurde erstmals 1953 von METROPOLIS et al. [MRR<sup>+</sup>53] als Algorithmus zur Erzeugung Boltzmann-verteilter Ensemble realisiert. KIRKPATRICK et al. [KGJV83] erweiterten 1983 den Metropolis Algorithmus zum *Simulated Annealing*, das in der Folgezeit zahlreiche Anwendung auf verschiedenste Optimierungsprobleme gefunden hat [LA87]. Der Metropolis Algorithmus ist gerade so konstruiert, daß als stationäre Verteilung die Boltzmann-Verteilung

$$P(x) = \lim_{t \rightarrow \infty} P(x, t) \sim e^{-\beta F(x)}$$

angenommen wird und so wollen wir Algorithmen mit einem solchen Verhalten *Boltzmann Strategien* nennen. Eine mögliche Realisierung in Form einer PDE ist durch

$$\frac{\partial}{\partial t} P(x, t) = \beta D \nabla_x (P \nabla_x F) + D \Delta_x P. \quad (3.1)$$

gegeben. Durch einen Ansatz der Form

$$P(\vec{x}, t) = \exp \left[ -\frac{\beta F(x)}{2} \right] y(\vec{x}, t) \quad (3.2)$$

erhält man zusammen mit der Trennung von Orts- und Zeitvariable die Eigenwertgleichung

$$H\psi(\vec{x}) = D\Delta\psi(\vec{x}) - V(\vec{x})\psi(\vec{x}) = -\epsilon\psi(\vec{x}) \quad (3.3)$$

mit dem Eigenwert  $\epsilon$  und einer transformierten Fitness

$$V(\vec{x}) = \frac{\beta^2}{4}D\nabla F \cdot \nabla F - \frac{\beta}{2}D\Delta F \quad (3.4)$$

was zur Lösung

$$P(\vec{x}, t) = \exp\left[-\frac{\beta}{2}F(\vec{x})\right] \sum_{i=0}^{\infty} c_i \psi_i(\vec{x}) \exp(-\epsilon_i t) \quad (3.5)$$

führt. Bei jeder Simulation eines solchen Algorithmus mittels eines Ensembles von Suchern kann der Mittelwert der Fitness  $\langle F \rangle$  ohne zusätzlichen Aufwand berechnet werden. Deshalb erscheint es natürlich, die folgende Geschwindigkeit

$$\begin{aligned} v^{(1)} &= -\frac{d}{dt}\langle F \rangle = D\beta\langle \nabla F \cdot \nabla F \rangle - D\langle \Delta F \rangle \\ \langle F \rangle &= \frac{\int P(x, t)F(x)dx}{\int P(x, t)dx} \end{aligned} \quad (3.6)$$

zu betrachten. Am Beispiel einer Parabel

$$F(x) = F_{min} + \frac{1}{2} \sum_{i=1}^d a_i (\vec{x}_i - \vec{x}_i^{(0)})^2 \quad (3.7)$$

konnte mittels der Anfangsverteilung  $P(x, 0) = \delta(x - x_0)$  (alle Sucher befinden sich im Punkt  $x_0$ ) eine geschlossene Formel für diese Geschwindigkeit

$$v^{(1)} = \sum_{i=1}^d a_i D(\beta a_i x_0^2 - 1) \exp(-a_i \beta D 2t) \quad (3.8)$$

angegeben werden. Desweiteren kann man die zeitliche Entwicklung des Mittelwertes  $\langle F \rangle$  als auch die Varianz der Fitness  $\sigma^2$  berechnen und erhält

$$\langle F \rangle = F_{min} + \sum_{i=1}^d \frac{1}{2\beta} (\beta a_i x_0^2 - 1) \exp(-a_i \beta D 2t) \quad (3.9)$$

$$\sigma^2 := \langle F^2 \rangle - \langle F \rangle^2 = \frac{1}{2\beta^2} \sum_{i=1}^d ((2x_0^2 \beta a_i - 1)(e^{-2Da_i\beta t} - e^{-4Da_i\beta t}) + 1 - e^{-2Da_i\beta t}) \quad (3.10)$$

Für die Simulation dieser Strategie verwenden wir die enge Beziehung zwischen der Fokker-Planck Gleichung (wie (3.1)) und der Langevin Gleichung. In diesem Fall verwenden wir die Gleichung

$$\frac{d}{dt} \vec{x} = -D\beta \nabla F + \sqrt{2D} \xi(t) \quad (3.11)$$

wobei  $\xi(t)$  Gaußsches weißes Rauschen,  $D$  die Diffusionskonstante und  $\beta$  die inverse Temperatur ist. Mit Hilfe der Diskretisierung

$$x(t+1) = x(t) + \sqrt{2D\Delta t} \xi(t) - D\beta \frac{F(x(t) + \Delta x) - F(x(t) - \Delta x)}{2\Delta x} \Delta t \quad (3.12)$$

wobei  $\Delta t$  der Zeitschritt und  $\Delta x$  die Boxlänge ist, wurde die Boltzmann Strategie (3.1) simuliert. Dieser Algorithmus zeigt für den Fall  $F(x) = x^2$  eine sehr gute Übereinstimmung mit der Theorie. In Bild 3.1 ist dieser Vergleich für den Mittelwert und die Varianz der Fitness zu ersehen. Die durchgezogene Linie kennzeichnet die analytische Lösung. Die Populationsgröße beträgt

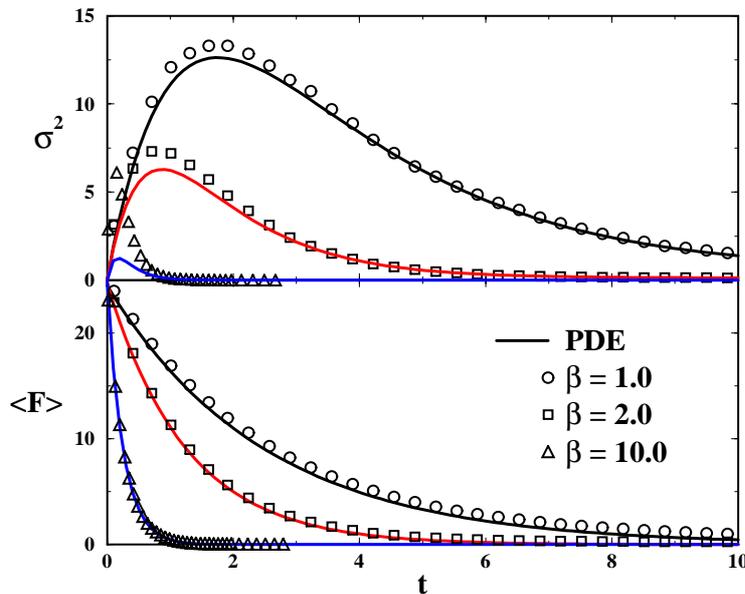


Abbildung 3.1: Mittelwert der Fitness und die Varianz der Fitness für die Parabel  $x^2$

$N = 10000$ . Weitere Simulationen und eine genauere Diskussion findet man im Artikel [AER97] sowie im nächsten Abschnitt.

**Ergebnis:**

Die Boltzmann Strategie ist schnell auf Landschaften mit langgezogenen Bergen. Im Fall von hohen Bergen versagt diese Strategie in Abhängigkeit von der Temperatur. Bei praktischen Problemen ist diese Strategie leicht einzusetzen.

## 3.2 Die Darwin Strategie

**(T. Aßelmeyer)**

Die zweite wichtige und - im eigentlichen Sinne - evolutionäre Strategie ist die Darwin Strategie. Wir verwenden dabei das Fisher-Eigen Modell der Evolution, das erstmals durch FISHER 1930 untersucht [Fis30] und von EIGEN als Standardmodell für Systeme mit Konkurrenz und Selektion formuliert wurde [Eig71]. Kennzeichnend für das Fisher-Eigen Modell ist eine fitnessproportionale Vermehrung der Individuen bei konstanter Populationsgröße [EF82]. Evolutionäre Strategien, die dies erfüllen, wollen wir *Darwin Strategien* nennen. Die Dynamik der Darwin Strategie ist gegeben durch

$$\frac{\partial}{\partial t} P(\vec{x}, t) = [\langle F \rangle - F(\vec{x})] P(\vec{x}, t) + D \Delta P(\vec{x}, t). \quad (3.13)$$

Mit Hilfe des Ansatzes

$$P(\vec{x}, t) = \exp \left[ \int_0^t \langle F \rangle (t') dt' \right] y(\vec{x}, t) \quad (3.14)$$

erhält man auf ähnliche Art und Weise die Eigenwertgleichung

$$-\epsilon_i \psi_i(\vec{x}) = D \Delta \psi_i(\vec{x}) - F(\vec{x}) \psi_i(\vec{x}) \quad (3.15)$$

und die Lösungen sind durch

$$y(\vec{x}, t) = \sum_i a_i e^{-\epsilon_i t} \psi_i(\vec{x}). \quad (3.16)$$

gegeben. Ein Vergleich zwischen (3.3) und (3.15) offenbart die tiefen Gemeinsamkeiten zwischen beiden Strategien. Die Boltzmann Strategie ist bzgl. der stationären Zustände äquivalent zur Darwin Strategie, falls man die Fitness durch (3.4) transformiert. Diese Strategie läßt sich auch in guter Näherung als (1 + 1) Strategie nach RECHENBERG und SCHWEFEL ansehen [Rec73b, Sch77b].

Nun diskutieren wir die Geschwindigkeiten auf eine ähnliche Art und Weise wie bei der Boltzmann Strategie. Dazu betrachten wir wieder die Geschwindigkeit

$$v^{(1)} = -\frac{d}{dt} \langle F \rangle = \langle F^2 \rangle - \langle F \rangle^2 - D \langle \Delta F \rangle . \quad (3.17)$$

und erhalten für die Fitness (3.7) durch dieselbe Anfangsverteilung  $P(x, u) = \delta(x - x_0)$  die folgende Formel für den Mittelwert der Fitness

$$\langle F \rangle = F_{min} + \sum_{i=1}^d \left( \frac{1}{2} \sqrt{2a_i D} \tanh(t\sqrt{2a_i D}) - \frac{a_i (x_0)_i^2}{2 \cosh^2(t\sqrt{2a_i D})} \right) \quad (3.18)$$

und für die Geschwindigkeit

$$v^{(1)} = \sum_{i=1}^d \sqrt{Da_i} \frac{\sqrt{2a_i} (x_0)_i^2 \sinh(t\sqrt{2Da_i}) - \sqrt{a_i D} \cosh(t\sqrt{2Da_i})}{\cosh^3(t\sqrt{2Da_i})} . \quad (3.19)$$

Die Simulation der Darwin Strategie wird auf die Simulation einer Mastergleichung zurückgeführt. Dies ist ein wohlbekanntes Problem aus der Theorie der Stochastischen Prozesse. Dazu wird die Wartezeitverteilung [Gil76, Fei77, Gil78, Str63] benutzt. Diese Methode findet man in der Literatur unter dem Begriff Gillespie-Algorithmus. Die Mutation wird auch durch Gaußsches weißes Rauschen simuliert, wobei ein besonderes Augenmerk auf die Zeitskalen gelegt wird. Selektionsprozesse werden in diesem Algorithmus durch Geburts- und Sterbeprozesse dargestellt. Da die ganze Prozedur relativ kompliziert ist, verweisen wir wieder auf den Artikel [AER97] in dem alle Einzelheiten zu finden sind. Für den Fall  $F(x) = x^2$  zeigt Bild 3.2 eine sehr gute Übereinstimmung zwischen Theorie und diesem Algorithmus für den Mittelwert der Fitness. Die durchgezogene Linie kennzeichnet die analytische Lösung. Entsprechende Simulationen für den Doppelpf  $F(x) = 0.16 + ((x/6)^2(x/6 - 0.9)(x/6 + 0.8))$  sowie eine genaue Beschreibung der verwendeten Algorithmen sind in [AER97] zu finden. Durch eine Approximation kann dieses Problem auch analytisch behandelt werden. Dazu betrachten wir die Fitnessfunktion im Bild 3.3, die einfach eine Zusammensetzung von 2 Parabeln ist. Berechnet man nun den Strom von Suchern, die durch den Punkt  $x = 0$  des Walls "hindurchströmen", d.h. berechnet man

$$\Phi(x, t) = \frac{\partial}{\partial t} \int_x^\infty P(y, t) dy \quad (3.20)$$

so erhält man für verschiedene Krümmungen der Parabeln die in Bild 3.4 gezeigten Ergebnisse für die Boltzmann und die Darwin Strategie. Das linke

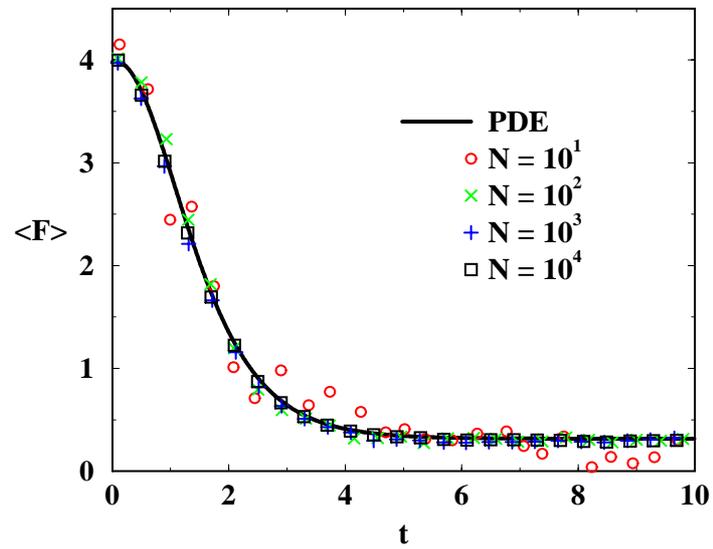


Abbildung 3.2: Mittelwert der Fitness für die Parabel  $x^2$

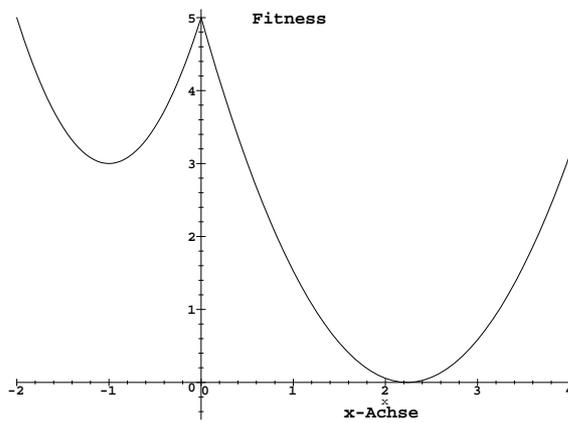


Abbildung 3.3: Fitnessfunktion für die einfachste Form eines Walls

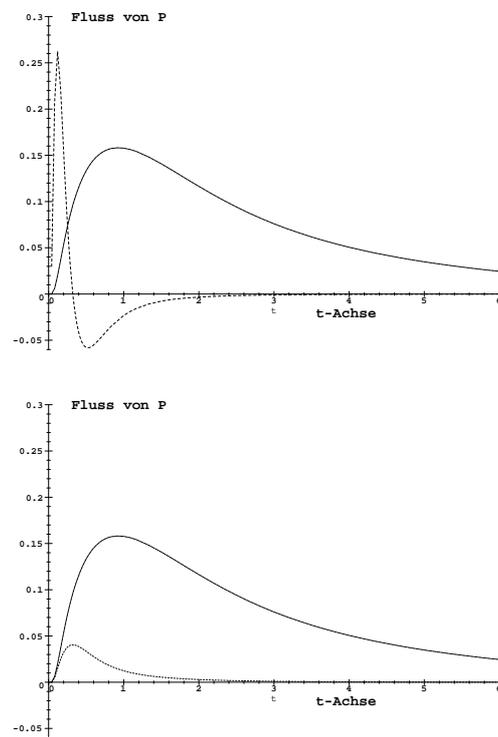


Abbildung 3.4: Strom für Boltzmann (links) und Darwin (rechts) in Bezug auf verschiedene Krümmungen des einfachen Walls

Bild zeigt das Verhalten der Boltzmann Strategie bei konstanter Temperatur für kleine Krümmung (volle Linie) und für große Krümmungen (gestrichelte Linie). Gestartet wird immer in der anderen Parabel bei  $x = -0.5$ . Die volle Linie zeigt, daß Boltzmann über den Wall “springt”, während die gestrichelte Linie die Umkehr des Stromes darstellt. Das bedeutet aber, daß die Strategie in diesem Falle nicht in endlicher Zeit über den Wall kommt. Ein ähnliches Ergebnis erhält man auch für die Darwin Strategie aber jetzt mit umgekehrtem Verhalten bzgl. der Krümmung. Das rechte Bild zeigt das entsprechende Verhalten, wobei die volle Linie den Strom für große Krümmungen und die andere Kurve den Strom für kleine Krümmungen darstellt. Der Fluß von einer Parabel zur anderen nimmt also mit fallender Krümmung auch ab. Damit verringert sich aber auch die Geschwindigkeit der Strategie. Wir erhalten damit folgendes Resultat:

**Ergebnis:**

Die Darwin Strategie ist schnell auf Landschaften mit steilen, aber engen Bergen und langsam auf Landschaften mit flachen, ausgedehnten Bergen. Für die Boltzmann Strategie erhält man das umgekehrte Verhalten.

Aufgrund dieses Resultates erscheint es nun plausibel beide Strategien zu mischen. An dieser Stelle scheint es angebracht eine Beziehung zur Klassifikation herzustellen. Dort wurde ausgesagt, daß die Katastrophen das Verhalten der Strategie bestimmen. Das stimmt natürlich nur für das Verhalten des Eigenwertproblems (z.B. (3.3)). Bezüglich dieses Eigenwertproblems unterschieden sich die Gleichungen um die Transformation (3.4) der Fitnessfunktion. Genau dieser Unterschied ist aber für den Unterschied zwischen der Boltzmann und der Darwin Strategie verantwortlich.

### 3.3 Gemischte Strategien

(T. Aßelmeyer)

Diese Strategie wurde durch Simulationen von Netzwerken gefunden [EE86, BEE87a, BE90]. Natürlich wurde schon vermutet, daß die gemischte Strategie oft schneller und robuster als die Boltzmann und die Darwin Strategie ist, aber ein direkter Beweis fehlte noch. Durch eine eingehende Analyse beider Strategien [AE97] konnte festgestellt werden, daß bezüglich einer geeignet definierten Geschwindigkeit beide Strategien ein gegenläufiges Verhalten aufweisen. Damit erscheint es jetzt sogar natürlich, eine Mischung beider Strategien anzusetzen.

Ein Vergleich zwischen den beschreibenden Gleichungen der Boltzmann

(3.1) und Darwin Strategie (3.13) zeigt den gemeinsamen Term  $D\Delta P$  beider Strategien. Deshalb erscheint es natürlich, beide Gleichungen einfach zu addieren, um so zur Beschreibung einer Gemischten Strategie zu kommen. Es ergibt sich dabei folgende Gleichung

$$\frac{\partial}{\partial t}P(\vec{x}, t) = (D_B + D_D)\Delta P(\vec{x}, t) + \beta D_B \nabla(P \nabla F) \quad (3.21)$$

$$+ \gamma[\langle F \rangle - F(\vec{x})]P(\vec{x}, t) . \quad (3.22)$$

mit den zwei Diffusionskonstanten  $D_B$  und  $D_D$ , die den jeweiligen Strategien zugeordnet sind. Die entsprechenden Grenzfälle ergeben sich durch  $\gamma = 0$  bzw.  $\beta = 0$ . Somit erhalten wir eine zwei-parametrische Familie von Strategien (siehe Bild 3.5. Wie können nun die Parameter angepaßt werden? In den

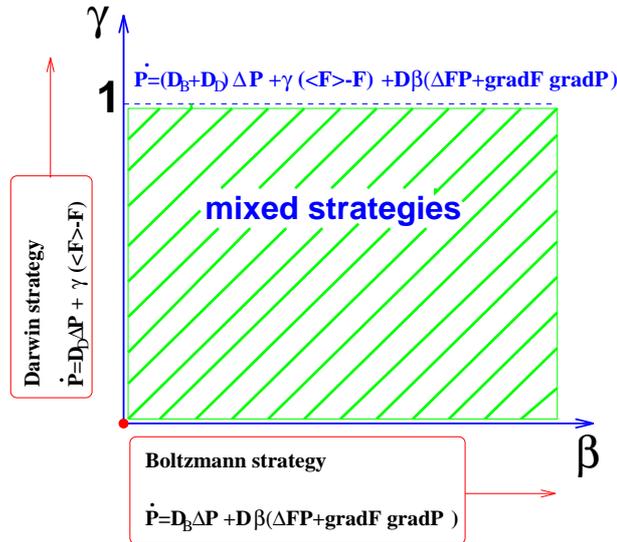


Abbildung 3.5: Darstellung der Gemischten Strategie als 2-parametrischen Familie

meisten Fällen werden die optimalen Parameter bei  $\gamma > 0$  und  $\beta > 0$  liegen. Berechnet man die oben definierte Geschwindigkeit  $v^{(1)}$  für diese Strategie, so erhält man

$$v^{(1)} = -\frac{d}{dt}\langle F \rangle = \gamma(\langle F^2 \rangle - \langle F \rangle^2) + D_B \beta \langle \nabla F \cdot \nabla F \rangle - (D_B + D_D)\langle \Delta F \rangle . \quad (3.23)$$

Daraus kann man unter der Annahme einer konstanten Geschwindigkeit  $v^{(1)} = v$  bei festem  $D$  und  $\beta$  den kritischen Wert von  $\gamma$  berechnen und

erhält

$$\gamma_{\text{Krit}} = \frac{(D_B + D_D)\langle \Delta F \rangle - D_B \beta \langle \nabla F \cdot \nabla F \rangle}{\langle F^2 \rangle - \langle F \rangle^2}. \quad (3.24)$$

Der wahre Wert von  $\gamma$  muß oberhalb des kritischen Wertes liegen. Durch viele Simulationen konnte die Verbesserung der Strategie gegenüber den reinen Strategien nachgewiesen werden. Die Simulation dieser Strategie erfolgt einfach durch die Mischung der Algorithmen zur Simulation der Boltzmann und Darwin Strategie. Wie oft selektiert wird, bestimmt allein der Parameter  $\gamma$ . Der Punkt 3.2 (Netzwerkoptimierung) gibt ein Beispiel für die erfolgreiche Anwendung der Gemischten Strategie auf ein praktisches Problem.

Ein Vergleich zwischen Boltzmann und der Gemischten Strategie für verschiedene Dimensionen und Populationsgrößen zeigt einen deutlichen Vorteil der Gemischten Strategie gegenüber den anderen Strategien. In Bild 3.6 kann man durch eine solche Simulation diesen Fakt gut erkennen. Nun läßt sich

Abbildung 3.6: Erreichte Fitness in Bezug auf die Populationsgröße für verschiedene Strategien, Fitness: Doppelpf

folgendes Ergebnis formulieren:

**Ergebnis:**

Die Gemischte Strategie ist eine Mischung zwischen einer Darwin und einer Boltzmann Strategie. Durch die Ergebnisse der vorherigen zwei Abschnitte wird nahegelegt, daß diese Strategie wirklich schneller sein sollte. Sowohl Simulationen als auch analytische Rechnungen bestätigen diese Vorstellung. Die Strategie wurde schon mehrfach praktisch eingesetzt und zeigt z.B. im Falle der Netzwerkoptimierung deutlich bessere Ergebnisse als die reinen Strategien und ist vor allem robuster.

### 3.4 Die adaptive Darwin Strategie

(T. Aßelmeyer)

Entsprechend SCHWEFEL [Sch81a] ist eine konstante Fortschrittsgeschwindigkeit der Evolutionsstrategie ein Nachteil, da die Strategie an Stellen, die weit vom Optimum entfernt sind, genauso schnell ist, wie an den interessanten Stellen nahe des Optimums. Die Lösung dieses Problems besteht darin,

daß man die Schrittweite  $\sigma$  als zusätzliche Variable betrachtet, die in den Selektionsprozeß einbezogen wird. In der bisherigen Darwin Strategie

$$\frac{\partial}{\partial t} P(\vec{x}, t) = D \Delta_x P + (\langle F \rangle(t) - F(\vec{x})) P$$

mit der Fitnessfunktion  $F$ , wurde die Schrittweite, ausgedrückt durch die Diffusionskonstante  $D$ , konstant gehalten. Nun erweitern wir den Suchraum  $X$  um eine Koordinate und es ergibt sich  $(\vec{x}, \sigma) \in X \times \mathbb{R}$ . Die Mutation der Schrittweite erfolgt durch eine log-normalverteilte Zufallszahl, die durch die Transformation  $\sigma = \exp(\eta)$  mit  $\eta$  als normalverteilte Zufallszahl dargestellt werden kann. Um die entsprechende Gleichung aus diesen Vorstellungen zu erhalten, werden wir auf einen von SCHIMANSKY-GEIER [SG96] entwickelten Formalismus zurückgreifen. Dazu betrachtet man zwei Sucher, die durch die Paare  $(x, \eta_x)$  und  $(y, \eta_y)$  beschrieben werden. Sei  $F(x)$  die zu minimierende Fitnessfunktion und  $f(x) = \langle F \rangle - F(x)$  die Fitness oberhalb des zeitabhängigen Mittelwertes. In einem Schritt des Algorithmus wird nun die Selektion für  $x$  mit der Rate  $\gamma$  und die für  $\eta_x$  mit der Rate  $\kappa$  entsprechend  $f(x)$  ausgeführt. Für die Selektion ergeben sich damit folgende Übergänge

$$(x, \eta_x, y, \eta_y) \begin{cases} \xrightarrow{\gamma |f(x)|} (x, \eta_x, x, \eta_y) & : f(x) > 0 \\ \xrightarrow{\gamma |f(x)|} (y, \eta_x, y, \eta_y) & : f(x) < 0 \\ \xrightarrow{\kappa |f(x)|} (x, \eta_x, y, \eta_x) & : f(x) > 0 \\ \xrightarrow{\kappa |f(x)|} (x, \eta_y, y, \eta_y) & : f(x) < 0 \end{cases} \quad (3.25)$$

Die ersten beiden Übergänge beschreiben die Selektion in  $x$ -Richtung während die nächsten zwei Übergänge die Selektion der Schrittweite bewirken. Um die zeitliche Änderung von  $P(x, t)$  durch die Übergänge (3.25) zu beschreiben, betrachtet man die Mastergleichung [vK81]. Dazu konstruiert man aus den Übergängen die entsprechenden Raten  $W(u, \acute{u}, v) := W((\acute{u}, v) \rightarrow (u, v))$  mit  $u = (x, \eta_x)$  und  $v = (y, \eta_y)$ . Dann ist die Mastergleichung gegeben durch

$$\begin{aligned} \frac{\partial}{\partial t} P(u, v, t) &= \int W(u, \acute{u}, v) P(\acute{u}, u, t) - W(\acute{u}, u, v) P(u, v, t) d\acute{u} \\ &+ \int W(v, \acute{v}, u) P(u, \acute{v}) - W(\acute{v}, v, u) P(u, v, t) d\acute{v}, \end{aligned} \quad (3.26)$$

wobei wir die Zweiteilchenverteilung  $P(u, v, t)$  zu betrachten haben, da unser Selektionsschema eine Wechselwirkung zwischen zwei Suchern darstellt. Die Einzelheiten der Herleitung sind an dieser Stelle nicht interessant. Mittels der Näherung der Korrelationsfreiheit  $P(x, y) = P(x) P(y)$  erhalten wir aus

obiger Gleichung das Ergebnis

$$\frac{\partial}{\partial t} P(x, \eta_x, t) = \kappa f(x) P(x, \eta_x, t) = \kappa (\langle F \rangle - F(x)) P(x, \eta_x, t) \quad (3.27)$$

$$\text{mit} \quad \langle F \rangle = \frac{\int F(x) P(x, \eta_x, t) dx d\eta_x}{\int P(x, \eta_x, t) dx d\eta_x} \quad (3.28)$$

Berücksichtigt man die Tatsache, daß die Rate der  $\eta_x$ -Selektion nicht von  $\eta_x$  explizit abhängt, dann erscheint dieses Ergebnis nicht erstaunlich. Im Gegensatz dazu ist der Einfluß der Selbstanpassung im Mutationsterm nicht zu vernachlässigen. Nimmt man an, daß die  $x$ -Mutation mit einer Normalverteilung mit Standardabweichung  $\exp(\eta)$  und die  $\eta$ -Mutation mit der Standardabweichung 1 erfolgt, so ergibt sich die Mastergleichung

$$\begin{aligned} \frac{\partial}{\partial t} P(x, \eta_x, t) &= \int (N(y - x, \exp(\eta_y)) N(\eta_y - \eta_x, 1) P(y, \eta_y, t) \\ &\quad - N(x - y, \exp(\eta_x)) N(\eta_x - \eta_y, 1) P(x, \eta_x, t)) dy d\eta_y \end{aligned} \quad (3.29)$$

mit  $N(x, \exp(\eta)) = (4\pi \exp(\eta))^{-1/2} \exp(-x^2 \exp(-\eta)/2)$ . Unter der Voraussetzung, daß die Strategie nur kleine Schritte macht und das sich dabei die Raten und die Verteilungsfunktionen nicht schlagartig ändern, werden die Funktionen im Integral entwickelt. Berücksichtigt man alle Terme bis zur zweiten Ordnung, dann erhält man für die Mutation

$$\frac{\partial}{\partial t} P(x, \eta_x, t) = \frac{1}{2} e^{\eta_x} \Delta_x P(x, \eta_x, t) + \frac{1}{2} \Delta_{\eta_x} P(x, \eta_x, t) . \quad (3.30)$$

Zusammen mit der Selektionsgleichung (3.27) erhalten wir die adaptive Darwin Strategie

$$\frac{\partial}{\partial t} P(x, \eta_x, t) = (\langle F \rangle - F(x)) P(x, \eta_x, t) + \frac{1}{2} e^{\eta_x} \Delta_x P(x, \eta_x, t) + \frac{1}{2} \Delta_{\eta_x} P(x, \eta_x, t) \quad (3.31)$$

welche wiederum als Eigenwertproblem

$$\frac{\partial}{\partial t} P = -HP \quad \text{mit} \quad H\Phi(\vec{x}, \eta_x) = \lambda\Phi(\vec{x}, \eta_x) \quad (3.32)$$

aufgefaßt werden kann. Zusammen mit den Eigenfunktionen  $\Phi_i(\vec{x}, \eta_x)$  und den Eigenwerten  $\lambda_i$  ergibt sich die allgemeine Lösung

$$P(x, \eta_x, t) = \sum_i c_i \exp(\lambda_i t) \Phi_i(\vec{x}, \eta_x) . \quad (3.33)$$

Die Konvergenz der obigen Reihe für alle Koeffizienten ist nur dann gesichert, wenn die Eigenwerte  $\lambda_i$  positiv sind. Die notwendige Bedingung ergibt sich zu

$$\lambda_i > 0 \quad \Leftrightarrow \quad F(x) - \langle F \rangle > 0 \quad (3.34)$$

Für lange Zeiten erzeugt der Diffusionsterm in  $\eta_x$ -Richtung eine vollständig ausgebreitete Verteilung, da der Selektionsterm nur auf die  $\vec{x}$ -Richtungen wirkt. Durch die Kopplung zwischen der Schrittweite (Diffusionskonstante) in  $\vec{x}$ -Richtung und der Koordinate  $\eta_x$  wird im Grenzfall  $t \rightarrow \infty$  die Verteilung in  $\eta_x$ -Richtung stärker ausgebreitet. Im Falle großer positiver  $\eta_x$ -Werte kann man den Term  $\langle F \rangle - F(x)$  in der Gleichung gegenüber der Diffusion vernachlässigen und man erhält dann auch eine ausgebreitete Lösung in  $\vec{x}$ -Richtung. Diese Argumentation zeigt, daß für große  $t$  die Lösung zu großen negativen  $\eta_x$ -Werten strebt, welches gleichbedeutend damit ist, daß sich die Verteilung lokalisiert ( $\eta_x$  ist die Standardabweichung der Verteilung über  $X$ ).

Um diese qualitative Diskussion zu untermauern, soll als ein Beispiel der unsymmetrische Doppelpopf  $x^4 - 0.5x^2 + 0.1x$  (siehe Bild 3.7) mit dem globalen Optimum bei  $x = -.544$  betrachtet werden. Mit Hilfe eines im Projekt

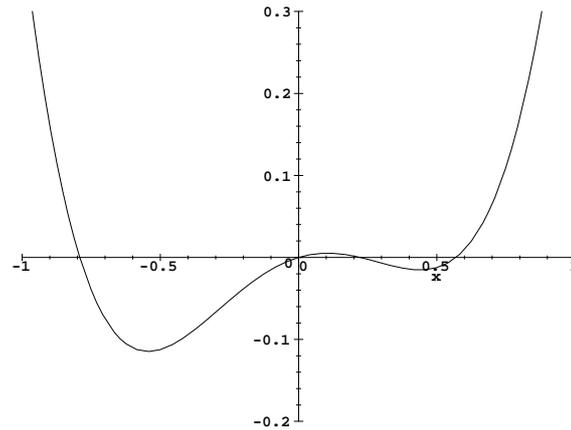


Abbildung 3.7: unsymmetrischer Doppelpopf

entwickelten Programmpakets (PDE2D) zur Integration einer solchen zweidimensionalen partiellen Differentialgleichung (PDE) wurde die entsprechende Gleichung

$$\frac{\partial}{\partial t} P = e^{\eta_x} \Delta_x P + \Delta_{\eta_x} P - (x^4 - 0.5x^2 + 0.1x)P \quad (3.35)$$

numerisch integriert. Gestartet wird mit einer Konfiguration in der alle Sucher an einem Punkt sind. In Bild 3.8 sieht man die entsprechende Folge von

Bildern, die die zeitliche Entwicklung der Population widerspiegeln. Dabei erkennt man recht gut das Anwachsen der Verteilung zu großen, negativen  $\eta_x$ -Werten (kleinen Schrittweiten) im globalen Optimum und das Anwachsen der Verteilung zu großen, positiven  $\eta_x$ -Werten (große Schrittweiten) im anderen lokalen Minimum. Dieses Verhalten würde man auch nach der Diskussion der Gleichung (3.31) erwarten. Andererseits folgt aus der Lokalisation der Verteilung über den lokalen Minima, daß diese Strategie die Tendenz zum Hängenbleiben aufweist. Wie man aus Erfahrung weiß, stimmt dieser Fakt mit der Wirklichkeit überein. Jetzt werden wir die Annahme machen, daß die Mutation von  $\eta_c$  nicht mit einer Normalverteilung durchgeführt wird. Dazu führen wir eine allgemeine Funktion der Form  $s(x, \eta_x, t)$  ein, mit

$$\frac{\partial}{\partial t} P(\vec{x}, \eta_x, t) = \Delta_{\eta_x} P + s(x, \eta_x, t) \Delta_x P + (\langle F(x) \rangle(t) - F(x)) P, \quad (3.36)$$

die in die Geschwindigkeit auf folgende Weise eingeht

$$v^{(1)} = -\frac{d}{dt} \langle F(x) \rangle = -\langle s(x, \eta_x, t) \Delta_x F(x) \rangle + \langle F(x)^2 \rangle - \langle F(x) \rangle^2. \quad (3.37)$$

Vom mathematischen Standpunkt her, ist die Geschwindigkeit ein Funktional von  $P$  und  $F$ . Daher braucht man nur die Geschwindigkeit nach diesen beiden Größen zu variieren, um die Geschwindigkeit  $v^{(1)}$  zu maximieren. Man erhält folgende Bedingungen:

$$\delta v^{(1)} = 0 \quad \text{und} \quad \delta^2 v^{(1)} < 0 \quad (3.38)$$

und daraus letztendlich

$$s(x, \eta_x, t) = \frac{F(x)(F(x) - 2\langle F(x) \rangle)}{\Delta_x F(x)} - P. \quad (3.39)$$

Diese Beziehung definiert eine neue Strategie, die durch die Gleichung

$$\begin{aligned} \frac{\partial}{\partial t} P(\vec{x}, \eta_x, t) &= \Delta_{\eta_x} P + \left( \frac{F(x)(F(x) - 2\langle F(x) \rangle)}{\Delta_x F(x)} - P \right) \Delta_x P \\ &+ (\langle F(x) \rangle(t) - F(x)) P \end{aligned} \quad (3.40)$$

beschrieben wird. Leider ist es bisher nicht gelungen einen brauchbaren Algorithmus zur effektiven Simulation dieser Gleichung zu finden. Trotzdem scheint es aber relativ klar zu sein, daß diese Strategie sowohl die Eigenschaften einer adaptiven Strategie besitzt als auch die Eigenschaft, in Landschaften nicht hängen-zubleiben. Damit ist aber noch nicht klar, ob diese Strategie das Optimum wirklich finden kann.

Zum Schluß sei noch kurz auf die Frage eingegangen, ob die adaptive Strategie beim Problem der Parabel wirklich schneller ist. Ausgehend von der Gleichung (3.31) wird daher für den 1-dimensionalen Fall  $F(x) = ax^2$  das Spektrum des stationären Operators  $H$  diskutiert werden, da für den Fall ohne Schrittweitensteuerung eine explizite Lösung bekannt ist.

Mittels einer Variablensubstitution  $u = x \exp(-\eta_x/2)$  erhält man aus (3.31) die Gleichung

$$\frac{\partial}{\partial t} P(u, t) = \left(1 + \frac{1}{4}u^2\right) \frac{d^2 P}{du^2} + \frac{u}{4} \frac{dP}{du} + \tilde{F}(u)P \quad (3.41)$$

und nach Trennung der Variablen die stationäre Eigenwertgleichung

$$\left(1 + \frac{1}{4}u^2\right) \frac{d^2 \Phi}{du^2} + \frac{u}{4} \frac{d\Phi}{du} + \tilde{F}(u)\Phi = -\lambda\Phi \quad . \quad (3.42)$$

Aufgrund der Linearität der obigen Gleichung wird ein Potenzreihenansatz für  $\Phi$  gewählt. Mit dem Potential  $\tilde{F}(u) = -ae^{\eta_x}u^2$  ergibt sich die folgende Gleichung für die Koeffizienten

$$a_{n+4} = \frac{4ae^{\eta_x}a_n - 4\lambda a_{n+2} + (n+2)^2 a_{n+2}}{4(n+3)(n+4)}$$

Konvergenz ist nur gesichert falls  $e^{\eta_x}$  beschränkt ist. Wie im vorherigen Abschnitt festgestellt wurde, strebt der Wert der  $\eta_x$ -Koordinate gegen  $-\infty$ . Damit kann aber der erste Term vernachlässigt werden und mit Hilfe der Abbruchbedingung zur Sicherung der Konvergenz dieser Reihe erhalten wir das Spektrum:

$$\lambda_k = \frac{k^2}{4}$$

Dieses Spektrum ist also im Gegensatz zum Fall ohne Schrittweitensteuerung, wo ein lineares Spektrum  $\lambda_k \sim k$  vorlag, quadratisch. Da in der Geschwindigkeit der Strategie die Form des Spektrums exponentiell eingeht, folgt daraus, daß die Strategie mit Schrittweitensteuerung auf der Parabel schneller als die gewöhnliche Strategie ist. Dieses Ergebnis wurde, wie oben beschrieben, durch Simulationen gestützt.

### Ergebnis:

Die Untersuchung einer Evolutionären Strategie nach SCHWEFEL konnte im konkreten Fall einer Fisher-Eigen Selektion explizit formuliert werden. Sowohl die Simulationen als auch die analytischen Rechnungen mit dieser

Gleichung zeigen die typischen Merkmale einer adaptiven Strategie. Im Fall der Parabel ist die adaptive Strategie schneller als die nicht-adaptive Strategie. Simulationen sowie auch mathematische Argumente zeigen aber, daß diese Strategie den Hang zum “hängenbleiben” besitzt.

### 3.5 Selfannealing

(H. Rosé)

Die Idee des Simulated Annealing besteht in der schrittweisen Abkühlung der Temperatur ( $\beta \rightarrow \infty$ ) des Metropolis Algorithmus, um die Suche auf Gebiete schon recht guter Fitneß zu konzentrieren. Bei zu schneller Abkühlung können jedoch lokale Minima in endlicher Zeit nicht verlassen werden, wohingegen bei zu langsamer Abkühlung keine Konzentration auf die Minima einsetzt. Dieses Problem der Wahl der geeigneten Abkühlung wurde von SALAMON und ANDRESEN im Rahmen thermodynamischer Überlegungen behandelt [NS88, AG93, AG94]. Ein ebensolches Problem stellt sich in der Wahl einer geeigneten Mutationsstärke. In den Evolutionsstrategien wird auch sie im Laufe der Suche optimiert [Sch81a]. Diese evolutionäre Form der Anpassung von Steuerparametern bietet den Vorteil, daß alle Parameter durch ein einheitliches und einfaches Prinzip adaptiert werden können. Durch seine Anwendung auf die gemischte Strategie wurde im Projekt eine neue evolutionäre Strategie entwickelt – die *Self Annealing Strategie* (SAS) – die erstere adaptive erweitert [Ros96a]. Sie zeigt als wichtigen neuen Effekt eine Selbstabkühlung der Temperatur und damit eine schnellere Konvergenz und optimalere Lösungen.

Die SAS wird ebenfalls durch ein Ensemble von Boltzmannsuchern realisiert, nur jeder besitzt nun ein eigenes  $\beta$  (inverse Temperatur), d.h. der Suchraum wird um eine Dimension erweitert  $X = \mathbb{R} \times \mathbb{R}$ . In einem Schritt des Algorithmus wird nun die Selektion für  $x$  mit der Rate  $\gamma$  und die für  $\beta_x$  mit der Rate  $\kappa$  für beide an Hand von  $f(x) = \langle F \rangle - F(x)$  vollzogen. Damit ergeben sich die Übergänge

$$(x, \beta_x, y, \beta_y) \left\{ \begin{array}{ll} \xrightarrow{\gamma |f(x)|} & (x, \beta_x, x, \beta_y) : f(x) > 0 \\ \xrightarrow{\gamma |f(x)|} & (y, \beta_x, y, \beta_y) : f(x) < 0 \\ \xrightarrow{\kappa |f(x)|} & (x, \beta_x, y, \beta_x) : f(x) > 0 \\ \xrightarrow{\kappa |f(x)|} & (x, \beta_y, y, \beta_y) : f(x) < 0 \\ \xrightarrow{\gamma \kappa f^2(x)} & (x, \beta_x, x, \beta_x) : f(x) > 0 \\ \xrightarrow{\gamma \kappa f^2(x)} & (y, \beta_y, y, \beta_y) : f(x) < 0 \end{array} \right. \quad (3.43)$$

Die ersten beiden entsprechen alleiniger  $x$ -Selektion, die nächsten alleiniger  $\beta_x$ -Selektion und die letzten der Selektion von  $x$  und  $\beta_x$  gemeinsam in einem Schritt. Die reine  $x$ -Selektion wird durch die Mastergleichung der gemischten Strategie beschrieben und da die Rate der reinen  $\beta_x$ -Selektion  $\kappa |f(x)|$  nicht von  $\beta_x$  abhängt, heben sich diese Übergänge einfach auf. Für die letzten beiden Übergänge erhalten wir aus der Mastergleichung den zusätzlichen Selektionsterm

$$\frac{\partial}{\partial t} P(u, t) = \gamma \kappa (f(u) |f(u)| + f(u) \langle |f| \rangle - \langle f |f| \rangle) P(u). \quad (3.44)$$

Damit ergibt sich die Dynamik der Self Annealing Strategie

$$\begin{aligned} \frac{\partial}{\partial t} P(x, \beta, t) &= \gamma f P + \gamma \kappa (f |f| + f \langle |f| \rangle - \langle f |f| \rangle) P \\ &+ \beta D \partial_x (P \partial_x f) + D \partial_x^2 P + D_\beta \partial_\beta^2 P. \end{aligned} \quad (3.45)$$

Die Self Annealing Strategie umfaßt alle drei oben behandelten Strategien und erweitert diese durch die Steuerung der Temperatur.

Die Frage ist nun: Besitzt die Strategie das gewünschte Verhalten der Selbstabkühlung der Temperatur  $T = 1/\beta$  in den Fitneßoptima? Um dies zu zeigen, wählen wir die recht schwierige Rastriginfunktion als Fitneß. Abb. 3.9 zeigt die Optimierung einer zweidimensionalen Rastriginfunktion durch den Self Annealing Algorithmus (implementiert in *ArtEvo*, s. 3.10.1). Die Fitnessfunktion (links) besitzt in  $x = (0, 0)$  ihr globales Minimum  $F = 0$ . Rechts ist die Entwicklung der mittleren Fitneß und der mittleren Temperatur wiedergegeben. Die bei  $x = (2, 2)$  gestartete Population passiert vier lokale Minima (vier Stufen in  $F_{min}$  - der Fitneß des Besten der Population), ohne stecken zu bleiben, da die Temperatur sich bis zum Zeitschritt  $t = 500$  kaum vermindert. Beginnend mit  $t = 700$  besetzt schließlich die gesamte Population das globale Minimum (starke Verringerung von  $\langle F \rangle$ ) und die Temperatur kühlt sich bis auf den minimal zugelassenen Wert von 0,1 ab.

Dieser Effekt der Selbstabkühlung kann wie folgt veranschaulicht werden: Sucher, mit hoher Temperatur, die sich bereits im Minimum befinden, sind noch zu beweglich. Sie verlassen das Minimum und werden ausselektiert, es sei denn sie gelangen in ein tieferes. Ist kein tieferes Minimum zu finden, so werden immer mehr Sucher hoher Temperatur ausselektiert – die Population kühlt sich im tiefsten zu erreichenden Minimum ab.

Abschließend wollen wir das Konvergenzverhalten aller vier Strategien für das Beispiel  $F(x) = x^2$  vergleichen. Dazu simulieren wir die Dynamik der Strategien mittels des im Projekt erstellten Programmes *ProEvo* (s. 3.10.1). Abb. 3.10 zeigt die, aus den numerischen Lösungen für gleiche Parameter gewonnene Entwicklung der mittleren Fitness und ihrer Standardabweichung

und Abb. 3.11 die Verteilungen nach zehn Sekunden ( $t = 10$ ). Die Boltzmann (BS) und die Darwin Strategie (DS) erreichen den annähernd gleichen Endwert, wobei DS sehr langsam startet, dann jedoch beschleunigt und BS erreicht. Die gemischte Strategie (BDS) kombiniert dieses Verhalten mit dem monotonen Absinken der BS und unterbietet damit beide. Die schnellste Suche und den besten Endwert zeigt die Self Annealing Strategie (SAS). Da sie die Temperatur erniedrigt, im Gegensatz zur Konstanz der drei anderen Strategien, beschleunigt sich nicht nur die Suche, sondern die Population konzentriert sich auch mehr im Minimum und erreicht daher einen tieferen Wert für  $\langle F \rangle$ . Diese Konzentration ist in ihrer Verteilung  $P(x, T, t = 10)$  in Abb. 3.11 rechts deutlich zu erkennen: Für kleine  $T$  schnürt sie sich um das Optimum  $x = 0$  zusammen. Die Verteilungen der anderen Strategien ist links wiedergegeben: Während BS und DS noch nicht ganz das Optimum erreicht haben, beginnt sich die BDS dort bereits zu konzentrieren.

Damit erhält man das Resultat, daß die SAS eine verbesserte Erweiterung der drei vorigen Strategien darstellt. Ihre adaptive Eigenschaften prädestinieren sie für praktische Anwendung auf stark multimodale Probleme. Dies zeigt das Beispiel der Rastriginfunktion (s. Abb. 3.11), in der die SAS ohne Steckenzubleiben das globale Optimum findet und erst dann die Temperatur gegen Null reduziert. Eine weiterführende eingehende Untersuchung dieser Strategie erscheint daher als sehr erfolgversprechend und notwendig.

## 3.6 Die Anwendung der Methode der adaptiven Suchräume

(T. Pöschel, V. Buchholtz)

### 3.6.1 Zusammenfassung

Es wurde eine neue Optimierungsstrategie eingeführt, die auf einer Glättung des Suchraumes durch Erhöhung der Dimension des Problems beruht. Ein kompliziertes Problem, wird so zunächst durch ein einfacheres ersetzt, indem die Dimension des Suchraumes vergrößert wird. Die Lösung des einfacheren Problems wird als Ausgangspunkt für die Lösung des komplizierteren Systems genutzt, indem nun schrittweise die Dimension des Suchraumes verringert wird, bis schließlich das Ausgangsproblem zu lösen ist.

### 3.6. DIE ANWENDUNG DER METHODE DER ADAPTIVEN SUCHRÄUME 65

Die Strategie wurde erfolgreich auf zwei Optimierungsprobleme angewandt.

Das erste Problem bestand darin, mit möglichst wenigen Wächtern einen kompliziert geformten Raum, der aus geradlinigen Wänden bestehen soll, vollständig zu überwachen. Es gibt eine Abschätzung für die maximal benötigte Anzahl von Wächtern, die besagt, daß man höchstens eine Anzahl von Wächtern benötigt, die einem Drittel der Anzahl der Wände entspricht. In den meisten Fällen genügt jedoch eine viel geringere Anzahl. Ziel der Optimierung war es, die Anzahl und die Position der Wächter zu bestimmen. Es stellte sich heraus, daß die benötigte Rechenzeit zur Lösung des Problems im Mittel wesentlich kleiner wurde, wenn man mit einer höheren Anzahl von Wächtern startet und die Zahl der Wächter nach und nach verringert, als wenn man gleich mit einer Wächterzahl startet, die in etwa der benötigten entspricht. Für das gewählte Problem ergab sich eine Verbesserung der Rechenzeit um einen Faktor 10. Die Strategie und die Ergebnisse wurden im *Int. J. of Bif. and Chaos* 7, S. 751 (1996) veröffentlicht. Der anschließende Abschnitt diskutiert diese ebenfalls im Detail.

Der Algorithmus wurde daraufhin verallgemeinert, indem in jedem Iterationsschritt zusätzlich zur Mutation der Orte der Wächter ein Hinzufügen oder Herausnehmen von Wächtern als Mutationsoperation erlaubt wurde. Um die Anzahl der Wächter in der Fitnessfunktion zu bewerten, wurde die ursprüngliche Funktion, die das Verhältnis aus eingesehener und gesamter Wandfläche war, um einen Term erweitert, der das Hinzufügen von Wächtern "bestraft" und das Herausnehmen "belohnt". Dieser Algorithmus besitzt hohe Stabilität (vergleichsweise kleine Fluktuationen der Rechenzeit) und führt zu geringen Rechenzeiten.

Da die Berechnung der Fitnessfunktion relativ aufwendig ist, wurde der Algorithmus parallelisiert. Die Parallelisierung erfolgte unter MPI. Die Individuen werden unter den vorhandenen Prozessoren aufgeteilt. Am Ende eines jeden Iterationsschrittes werden die berechneten Fitnesswerte verglichen, und die besten Individuen an die Prozessoren versandt.

In einer zweiten Implementierung wurde die Strategie zur Lösung eines Überdeckungsproblems angewandt. Eine beliebig geformte Grundfläche soll mit möglichst wenig identischen Rechtecken überdeckt werden. Auch in diesem Fall erweist sich die vorgeschlagene Strategie als erfolgreich. Startet man mit einer deutlich zu hohen Anzahl von Rechtecken, wird die Lösung zuverlässig in relativ geringer Rechenzeit gefunden. Verringert man die anfängliche Anzahl von Rechtecken, wird zunächst die Rechenzeit länger. Verringert man die Anzahl weiter, ist der evolutionäre Algorithmus nicht mehr zuverlässig in der Lage, die Lösung in angemessener Rechenzeit überhaupt zu finden obwohl die Zahl der beteiligten Rechtecke noch immer größer ist

als die Zahl der benötigten Rechtecke.

### 3.6.2 Adaptive evolutionäre Optimierung – Konvergenzbeschleunigung durch Erweiterung des Zustandsraumes

#### Einführung

Von vielen Autoren wurde gezeigt (z.B. [Gol89a]), daß Evolutionsalgorithmen unter bestimmten Voraussetzungen zur Lösung komplexer Optimierungsaufgaben geeignet sind. Es existiert eine weit entwickelte Theorie der Evolutionsprozesse (z.B. [EEF90a]) und es gibt eine große Zahl von „Rezepten“ für die erfolgreiche Anwendung evolutionäre Algorithmen (z.B. [Sch95]). Insbesondere für kombinatorische Optimierungsaufgaben, bei denen viele Standardalgorithmen versagen, konnten unter Verwendung von Evolutionsstrategien erhebliche Fortschritte erzielt werden.

Offensichtlich könnte jedes einzelne Optimierungsproblem effektiver (d.h. weniger rechenzeitaufwendig) mit einem deterministischen Algorithmus gelöst werden als mit einem Evolutionsalgorithmus, vorausgesetzt der deterministische Algorithmus ist bekannt. Der Vorzug von Evolutionsalgorithmen besteht in deren Einfachheit und Universalität. Um einen Evolutionsalgorithmus sinnvoll anwenden zu können, müssen zwei wichtige Voraussetzungen gesichert sein:

1. Es wird eine Fitness-Funktion benötigt, die eine Lösung einer Optimierungsaufgabe bewertet. Diese Fitness-Funktion muß äußerst effektiv berechenbar sein, d.h. ihre Auswertung darf nur wenig Rechenzeit benötigen, da sie während der Optimierungsprozedur extrem häufig auszuwerten ist. Folglich muß die Fitness-Funktion einfach sein.
2. Es wird ein Mutationsoperator benötigt, der die Topologie der Fitness-Funktion über dem Raum der zu optimierenden Parameter berücksichtigt. Falls der Mutationsoperator die Topologie nicht berücksichtigt, so kommt es vor, daß selbst kleine Mutationen zu extremen Veränderungen der Fitness führen. In solchen Fällen verwandelt sich der Evolutionsalgorithmus in stochastische Suche<sup>1</sup>.

---

<sup>1</sup>Genetische Programmierung [Gol89a] ist solch ein Beispiel. Jedes Rechnerprogramm besteht aus Konstanten und Anweisungen. Mindestens in Bezug auf die Anweisungen ist die Bedingung der starken Kausalität nicht gegeben: das Resultat eines Programms, das in einer gewöhnlichen Programmiersprache geschrieben ist, wird sich in der Regel extrem ändern, wenn ein Befehl (z.B. GOTO 100) durch einen anderen (z.B. write(\*,\*) 'HELLO WORLD') ersetzt wird. Die Hauptaufgabe der genetischen Programmierung muß unseres

### 3.6. DIE ANWENDUNG DER METHODE DER ADAPTIVEN SUCHRÄUME 67

In der Literatur können viele Beispiele gefunden werden, wo eine dieser beiden Voraussetzungen nicht gegeben ist, und wo Evolutionsalgorithmen demzufolge nicht effektiv arbeiten können.

Eine weitere sehr wichtige Eigenschaft von Evolutionsalgorithmen ist ihre triviale rechen-technische Parallelisierbarkeit: man teilt einfach die Individuen unter den parallel arbeitenden Prozessoren auf und vermeidet so unnötige Kommunikation zwischen den Prozessoren. Lediglich beim Selektions- und Rekombinationsschritt müssen wenige Daten ausgetauscht werden. Die Parallelisierbarkeit ist ein weiterer wichtiger Vorteil von Evolutionsalgorithmen gegenüber deterministischen Algorithmen, die häufig nicht oder nicht einfach effektiv parallelisierbar sind.

Eines der bekanntesten Beispiele, das das Interesse von Wissenschaftlern sehr verschiedener Fachrichtungen auf sich gezogen hat, ist das Problem des reisenden Händlers. Das Problem besteht darin, die kürzeste Rundreise zwischen  $N$  Städten zu finden. Es ist bewiesen, daß dieses Problem  $NP$ -vollständig ist [GJ79], d.h. daß kein deterministischer Algorithmus existieren kann, der dieses Problem in einer Zeit löst, die polynomial mit der Systemgröße  $N$  wächst. Viele Autoren betrachteten dieses Problem als Herausforderung, die Leistungsfähigkeit verschiedener Typen evolutionärer Strategien, als auch anderer Methoden zu testen (z.B. [Rei, BEE87b, BE88, HT85, WK88, WSS91]).

Im vorliegenden Abschnitt wollen wir eine Optimierungsaufgabe vorstellen, bei der es darum geht, daß jedes Individuum eine Mannschaft von Spielern ist, die eine vorgegebene Aufgabe durch kooperatives Verhalten gemeinsam lösen. Am Anfang ist dabei weder die Anzahl von Spielern pro Mannschaft  $N_{min}$ , die mindestens nötig ist, um die Aufgabe zu lösen, noch die detaillierte Lösung bekannt. Beide Antworten sollen m.H. des Evolutionsalgorithmus gefunden werden. Formal kann man die Lösung als Unterraum des Konfigurationsraumes der Dimension  $2 N_{min}$  auffassen.

Wir wollen weiterhin zeigen, daß selbst dann, wenn wir die benötigte Anzahl von Spielern  $N_{min}$  zur Lösung der Aufgabe kennen, es dennoch vorteilhaft sein kann, die Lösung zu finden, indem man zuerst die Lösung eines einfacheren Problems sucht, d.h. die Aufgabe mit  $N > N_{min}$  Spielern löst, und dann von dieser Lösung ausgehend  $N$  schrittweise verkleinert. Das Lösen einer solchen Hierarchie von Problemen für  $N_{min} + K$ ,  $N_{min} + K - 1$ ,  $\dots$ ,  $N_{min} + 1$  und schließlich für  $N_{min}$  kann wesentlich effektiver sein, als das direkte Lösen der Aufgabe für  $N_{min}$  Spieler.

---

Erachtens darin bestehen kausal starke Mutationsoperatoren zu entwickeln.

### Die Beschreibung des Problems

Angenommen, die  $M$  Wände eines Raumes mit polygonaler Grundfläche sollen mit  $N$  punktförmigen Lampen vollständig ausgeleuchtet werden. Die Frage, die uns hier interessieren soll, ist: *Wieviele Lampen sind mindestens nötig, die Wände des Raumes zu beleuchten, und wo müssen sie angebracht werden?*

Obwohl wir keinen Beweis für  $NP$ -Vollständigkeit des Problems liefern können, wird man wahrscheinlich vermuten, daß dieses Problem im allgemeinen Fall schwierig ist. Für den Fall, daß der Raum keine Säulen enthält, existiert ein Beweis für die obere Grenze für die Zahl  $N$  von Lampen, die notwendig ist, um die  $M$  Wände des Raumes zu beleuchten [Fis78]. In diesem Fall benötigt man höchstens  $N = \lfloor M/3 \rfloor$  Lampen, wobei  $\lfloor a \rfloor$  den ganzzahligen Anteil von  $a$  bezeichnet. Abbildung 3.12 zeigt den Grundriß eines Raumes, zu dessen Beleuchtung tatsächlich  $N = \lfloor M/3 \rfloor$  Lampen nötig sind. In den allermeisten Fällen werden jedoch wesentlich weniger als  $N = \lfloor M/3 \rfloor$  Lampen benötigt. Falls der Raum  $S$  Säulen polygonaler Querschnittsfläche enthält, so wird vermutet ([O'R87]), daß maximal  $N = \lfloor (M + S)/3 \rfloor$  Lampen benötigt werden, wobei  $M$  die Summe der Wände des Raumes und der Säulen bezeichnet. Dafür gibt es jedoch bis jetzt noch keinen Beweis.

Ein verwandtes Problem ist das sogenannte „Galeriewächter-Problem“ (Gallery Watchmen Problem) [Chv75]. Dabei ist die Frage zu beantworten, wieviele (unbewegliche) Wächter mindestens nötig sind, um alle Wände eines Museums mit polygonalem Grundriß beobachten zu können (z.B. [Ste94]), bzw. in der dynamischen Formulierung: wie verläuft der kürzeste Pfad eines Wachmanns um alle Punkte zu passieren von denen aus man insgesamt alle Wände des Museums einsehen kann? Es existieren viele verschiedene Formulierungen dieses Problems und viel theoretische Arbeit ist auf diesem Gebiet geleistet worden (z.B. [O'R87, Nta86, CN88, Nta92, NC91]). Eine technologisch interessante eindimensionale Variante eines dynamischen Wachmann-Problems wurde m.H. eines Evolutionsalgorithmus von Heckman [Hec95] untersucht. Eine Maschine besteht aus einer Anzahl von Greifern, die an einer geraden Stange befestigt worden sind, die quer über einem Förderband hängt. Die Steuerung der Maschine wurde optimiert, um Gegenstände zu ergreifen, die auf diesem Förderband transportiert werden. Der Algorithmus sollte entscheiden welcher der Greifer das nächste ankommende Teil aufnehmen soll.

Wir wollen annehmen, daß der Raum keine Säulen besitzt, deshalb ist die Anzahl der benötigten Lampen  $N_{min}$  immer kleiner oder gleich einem Drittel der Zahl der Wände  $N_{min} \leq \lfloor M/3 \rfloor$ . Die Fitness  $F$  eines Individuums, d.h. eines Sets von Lampen, wird durch die beleuchtete Fläche aller Wände

### 3.6. DIE ANWENDUNG DER METHODE DER ADAPTIVEN SUCHRÄUME 69

bestimmt:

$$F_i = \frac{\sum_{i=1}^M k_i}{\sum_{i=1}^M l_i}, \quad (3.46)$$

wobei  $k_i$  die beleuchtete Fläche der  $i$ -ten Wand, und  $l_i$  die Gesamtfläche dieser Wand ist. Die Fitness liegt also im Intervall  $F \in [0, 1]$ . Im Verlauf der Optimierung versuchen wir,  $F$  zu maximieren, indem wir die Positionen der Lampen verändern. Die Lösung ist dann gefunden, wenn wir die minimale Anzahl der Lampen bestimmt haben, die benötigt werden, den Raum zu beleuchten, d.h.  $F = 1$ . Die Funktion  $F$  ist also in den  $2N$  dimensionalen Raum der Koordinaten der  $N$  Lampen eingebettet. Diese Funktion kann eine komplexe Topologie aufweisen, insbesondere Unstetigkeitsstellen, lokale Extrema und flache langgestreckte Plateaus. Eine einfache Gradientenstrategie würde also mit Sicherheit versagen, das Optimum zu finden, und man muß folglich ein anderes Optimierungsverfahren anwenden.

Wir wollen die Optimierung unter Verwendung eines Evolutionsalgorithmus untersuchen, der in der Klassifizierung nach Schwefel [Sch81b, Sch95] vom Typ  $(\mu, \lambda)$  ist. Jedes Individuum  $\alpha$ , ( $\alpha = 1 \dots \mu$ ) in unserem Evolutionsspiel ist ein Set von  $N_{max} \geq N \geq N_{min}$  Lampen, wobei  $N_{max} = \lfloor M/3 \rfloor$  and  $N_{min}$  die (unbekannte) minimale Anzahl von Lampen ist, die benötigt werden, um den Raum zu beleuchten. Der Algorithmus startet mit  $\mu$  solcher Individuen, die Koordinaten der Positionen  $\vec{S}_i^\alpha$ ,  $i = 1 \dots N$  der Lampen werden zufällig gewählt. Die Optimierungsprozedur läuft dann folgendermaßen ab:

1. Die Individuen werden mutiert, indem die Positionen aller Lampen verändert wird:

$$\vec{S}_i^\alpha \rightarrow \vec{S}_i^\alpha + m_w \cdot \vec{A}_i^\alpha, \quad (3.47)$$

wobei  $i \in [1, N]$  und  $\alpha \in [1, \mu]$ . Die Komponenten des zweidimensionalen Vektors  $\vec{A}_i^\alpha$  werden zufällig aus dem Intervall  $[0, 1]$  gewählt. Die Mutationsschrittweite  $m_w$  ist konstant. Um zu verhindern, daß die Individuen in tiefen lokalen Maxima steckenbleiben, wird die Position einer Lampe mit einer sehr kleinen Wahrscheinlichkeit  $P$  vollständig zufällig bestimmt.

2. Alle Individuen werden bezüglich der Fitness-Funktion (3.46) bewertet.
3. Falls kein Set  $\alpha \in [1 \dots \mu]$  das Problem löst, d.h.  $F_\alpha \neq 1$ , so werden die  $\lambda$  Sets (Individuen) von Lampen mit den größten Fitness-Werten kopiert und in die nächste Generation übernommen. Die verbleibenden

$\mu - \lambda$  Individuen sterben aus. Die Schritte 1 bis 3 werden solange wiederholt, bis wenigstens eines der Individuen die Fitness  $F_\alpha = 1$  aufweist.

4. Falls wenigstens eines der Individuen die Fitness  $F = 1$  hat, so ist die Lösung des Problems gefunden und wir beginnen ausgehend von der gefundenen Lösung das nächstschwierigere Problem zu lösen, d.h. wir versuchen, dasselbe Optimierungsproblem zu lösen, wobei aber nun jedes Individuum nur mehr aus  $N - 1$  Lampen besteht. Die Lösung des vorangegangenen Problems mit  $N$  Lampen dient zur Initialisierung der neuen Positionen der  $N - 1$  Lampen pro Individuum: Das Gewinner-Individuum mit der Fitness  $F = 1$  wird  $\mu$  mal kopiert, und bei jedem der neuen Individuen wird eine zufällig gewählte Lampe entfernt. Danach starten wir die neue Optimierungsaufgabe mit  $N - 1$  Lampen, beginnend mit Schritt 1.

Die beschriebene Prozedur wird ausgeführt, bis keine neue Lösung mehr gefunden werden kann.

### Ergebnisse und Diskussion

Der vorgestellte Algorithmus wurde auf das Problem, einen Raum mit  $M = 82$  Wänden und dem Grundriß nach Abbildung 3.13 zu beleuchten. Die Lösung, in Abbildung 3.13 mit  $N_{min} = 10$  Kreuzen (genaugenommen 9 Kreuzen und einem Stern) dargestellt wurde mit einer evolutionären Optimierung mit  $\mu = 60$  Individuen gefunden. Die Optimierungsparameter waren  $\lambda = 5$ ,  $P = 10^{-3}$  und  $m_w = \min(x_{max}, y_{max}) \cdot 10^{-2}$ , wobei  $x_{max}$  und  $y_{max}$  die maximale Ausdehnung des Raumes in  $x$ - und  $y$ -Richtung bezeichnen.

In Abbildung 3.14 ist die Fitness  $F_I$  des besten Individuums  $I$  über der Anzahl der benötigten Evolutionsschritte aufgetragen. Dabei wurde mit einem Set von  $N = N_{min} = 10$  zufällig im Raum verteilten Lampen gestartet. Die Fitness wächst nicht monoton, sondern sie weist lange Phasen der Stagnation auf, die von plötzlichen Sprüngen unterbrochen werden. (Die Abszisse ist logarithmisch aufgetragen!) Dieses Verhalten, das für evolutionäre Algorithmen typisch zu sein scheint, wurde von mehreren Autoren bei der Optimierung verschiedener Probleme beobachtet, z.B. [Rec73a], und theoretisch begründet (s. [EEF90a]).

Normalerweise kennen wir *a priori* die Anzahl  $N_{min}$  von Lampen nicht, die zur Beleuchtung des Raumes benötigt werden, und wir starten die Optimierung mit  $N_0 > N_{min}$  Lampen pro Individuum. Wir werden zeigen, daß selbst dann, wenn wir diese Zahl kennen, es vorteilhaft wäre, zunächst das Optimum für eine größere Anzahl von Lampen zu suchen. Wir wollen

### 3.6. DIE ANWENDUNG DER METHODE DER ADAPTIVEN SUCHRÄUME 71

nun demonstrieren und begründen, daß die Optimierung in diesem Falle viel schneller verläuft. In unserem Beispiel des Raumes nach Abbildung 3.13 fanden wir die Lösung bis zu etwa zehnmals schneller.

Um die Daten fair vergleichen zu können, müssen wir sicherstellen, daß in jedem Evolutionsschritt etwa dieselbe Rechenzeit gerechnet wird, unbeeinflusst von der Anzahl von Lampen, aus denen die Sets bestehen. Deshalb wählten wir die Anzahl der Individuen  $\mu$  nach

$$\mu = \mu_{max} \cdot \frac{N_{min}}{N_0}, \quad (3.48)$$

wobei  $\mu_{max} = 60$  die Anzahl der Individuen einer Optimierung, die mit  $N_{min}$  Lampen pro Individuum ausgeführt wird. Folglich können wir die Anzahl der Evolutionsschritte mit der Rechenzeit identifizieren, wenn wir annehmen, daß die Rechenzeit hauptsächlich zur Auswertung der Fitness-Funktion verwandt wird, d.h. wenn die Rechenzeit ungefähr proportional der Gesamtzahl der Lampen ist.

Abbildung 3.15 zeigt, daß der Algorithmus im Mittel etwa zehnmals schneller ist, wenn mit Sets startet, die aus  $N_0 = 16$  Lampen bestehen, anstelle der mindestens erforderlichen  $N_{min} = 10$  Lampen. Jeder Datenpunkt in dieser Abbildung zeigt den Mittelwert der benötigten Optimierungsschritte, gemittelt über fünf Läufe mit verschiedenen Anfangskonfigurationen. Im Bereich  $N_{min} \leq N_0 \leq 16$  fällt die Gesamtrechenzeit, wenn die Zahl der Lampen pro Set zu Beginn der Simulation wächst. Starten wir mit  $N_0 > 16$  so wächst die Rechenzeit langsam. Für dieses Verhalten sind zwei Mechanismen verantwortlich:

1. Angenommen, wir suchen das Optimum durch stochastische Suche und angenommen, es existiert eine einzige Lösung des Optimierungsproblems mit  $N_{min}$  Lampen. Wenn wir die Zeit, die ein einzelner Sucher benötigt, um *eine bestimmte* der  $N_{min}$  Positionen zu finden, mit  $\tau$  bezeichnen, so benötigen  $N$  Sucher die Zeit

$$\tau^* = \frac{\tau}{NN_{min}}, \quad (3.49)$$

um *irgendeine* der Positionen zu finden. Nehmen wir weiter an, daß ein Sucher, der „seinen“ Platz gefunden hat, an diesem Platz verharret, bis die Lösung des Problems gefunden ist, so erhalten wir die Zeit, die benötigt wird, um *alle* Positionen zu finden, d.h. um das Problem zu lösen:

$$T(N) = \sum_{i=0}^{N_{min}-1} \frac{\tau}{(N-i)(N_{min}-i)}. \quad (3.50)$$

Ersetzen wir die Summe durch ein Integral, so finden wir

$$\begin{aligned} T(N) &\approx \tau \int_{\frac{1}{2}}^{N_{min}-\frac{1}{2}} \frac{1}{(N-x)(N_{min}-x)} dx \\ &= \frac{\tau \ln\left(\frac{2N-1}{(2N_{min}-1)(2N-2N_{min}+1)}\right)}{N_{min}-N}. \end{aligned} \quad (3.51)$$

Offensichtlich ist  $T(N) - T(N+1)$  positiv, d.h. die triviale Aussage, daß die Lösung mit  $N+1$  stochastischen Suchern schneller gefunden wird als mit  $N$ , wird korrekt wiedergegeben.

Typischerweise sind die „Attraktorgebiete“ der Positionen keinesfalls gleich groß, sondern ihre Größe variiert stark. Die Positionen mit den kleinsten „Attraktoren“ werden meist am Ende der Optimierung gefunden, und folglich wird die Rechenzeit stark von den letzten Termen der Summe (3.50) bestimmt.

Abbildung 3.16 zeigt die Resultate der Optimierung, sowie die diskutierten Abschätzungen.

- Abbildung 3.15 zeigt, daß die Rechenzeit für große  $N_0$  schwach wächst. Vergleichen wir die gefundenen Lösungen für verschiedene Set-Größen  $N$ , so finden wir, daß für  $N_0 \lesssim 18$  die Anfangslösung (für  $N = N_0$ ) häufig sehr dicht bei der endgültigen Lösung (für  $N = N_{min}$ ) liegt. Folglich kann die Lösung für  $N-1$  Lampen oft sehr schnell gefunden werden, indem man aus der schon bekannten Lösung für  $N$  Lampen einfach eine (die richtige!) Lampe fortnimmt und einige wenige Evolutionsschritte ausführt (s. Abbildung 3.17).

Für  $N_0 \gtrsim 18$  unterscheiden sich die Anfangslösungen oft sehr stark von der endgültigen Lösung, und folglich weichen die Lösungen der „benachbarten“ Probleme für  $N_0, N_0-1, \dots, N_{min}$  stark voneinander ab. Die Lösungen für große Lampenzahlen  $N$  sind nicht wirklich an die Geometrie des Raumes angepaßt, sondern sie sind eher das Resultat einer zufälligen Suche. Insofern ist die Lösung für  $N+1$  Lampen kaum hilfreich für das Auffinden der Lösung mit  $N$  Lampen. Folglich wächst die Rechenzeit in Abbildung 3.15 etwas für große Werte von  $N$ . Nichtsdestotrotz ist der Algorithmus noch immer viel schneller, als wenn man die evolutionäre Suche direkt mit  $N = N_{min}$  Lampen startete.

Abbildung 3.17 zeigt den Fortschritt der Optimierungsstrategie für  $N_0 = \{16, 18, 20, 22\}$  Lampen pro Set in der Startkonfiguration. Die

### 3.6. DIE ANWENDUNG DER METHODE DER ADAPTIVEN SUCHRÄUME 73

Abbildung sollte von rechts nach links gelesen werden: Die Ordinate zeigt die Anzahl von Evolutionsschritten, die notwendig waren, das Problem für  $N$  Lampen zu lösen. Im Falle des Laufes mit  $N_0 = 18$  Lampen wurde die endgültige Lösung für  $N = N_{min}$  fast sofort durch wegnehmen von überflüssigen Lampen gelöst.

Die Situation erinnert etwas an das Lernverhalten eines neuronalen Netzwerkes, das auf ein Zeichenerkennungsproblem angewandt wird: Wenn das Netzwerk aus zu vielen Neuronen besteht, so wird das Erkennungsproblem gelöst, indem jedes der Muster „auswendig“ gelernt wird, anstatt die Muster hinsichtlich charakteristischer Eigenschaften zu unterscheiden. Ein solches Netzwerk versagt, falls es ein unbekanntes, aber einem der gespeicherten Muster sehr ähnliches Zeichen erkennen soll. Ein kleines Netzwerk, das gerade so groß ist, daß es die Erkennungsaufgabe erlernen kann, abstrahiert u.U. sehr gut, lernt allerdings sehr langsam. Dazwischen liegt oft ein optimaler Kompromiß für die Anzahl der Neuronen des Netzwerkes.

#### **Zusammenfassung**

Wir haben einen Evolutionsalgorithmus auf ein komplexes Optimierungsproblem angewandt und dessen Lösungsverhalten studiert, bei dem jedes Individuum eine „Mannschaft“ ist, die durch kooperatives Verhalten eine Aufgabe lösen sollten. Sowohl die „Mannschaftsstärke“  $N$ , als auch die Lösung selbst waren zu Beginn der Optimierung unbekannt. Das Problem galt als gelöst, wenn die kleinste mögliche Mannschaft  $N_{min}$  gefunden war, für die die Fitness-Funktion ihren maximalen Wert annimmt  $F = 1$ .

Wir haben gezeigt, daß es für das untersuchte Problem vorteilhaft war, die Optimierung mit einer größeren Zahl zu starten, als letztendlich zur Lösung minimal nötig war  $N_0 > N_{min}$ , und dann ausgehend von der Lösung für  $N_0$  die Lösungen für  $N_0 - 1 \dots N_{min}$  zu suchen, wobei die Lösung für  $N + 1$  als Anfangskonfiguration für die Aufgabe mit  $N$  benutzt wurde. Wir fanden, daß es in unserem Beispiel bis zu zehnmal weniger Rechenzeit erforderte, die Kette der Aufgabe mit  $N_0, N_0 - 1, N_0 - 2, \dots, N_{min}$  zu lösen, als sofort mit der Lösung für  $N_{min}$  zu beginnen. Formal betrachtet lösen wir die Aufgabe zunächst in einem hochdimensionalen Raum mit Dimension  $2^N$ . Wenn diese Lösung gefunden ist, so begeben wir uns schrittweise in den Raum mit  $2^{N_{min}}$  Dimensionen.

Die theoretische Grundlage der Geschwindigkeitssteigerung in höherer Dimension ist durch die Morse-Theorie gegeben [Mil63]. Das Morse-Lemma sagt aus, daß (unter relativ schwachen Bedingungen) der Anteil der Sattelpunkte einer Funktion an der Gesamtzahl seiner kritischen Punkte mit der

Dimension wächst. Folglich fällt das Verhältnis von Extrema und Sattelpunkten mit wachsender Dimension. Die Bedeutung des Lemmas wird sehr anschaulich in einer und zwei Dimensionen. Um von einem Punkt aus das globale Maximum einer Funktion zu erreichen, so muß man in einer Dimension durch alle lokalen Extrema zwischen dem Startpunkt und dem Maximum „hindurchlaufen“. In zwei Dimensionen kann man u.U. um die lokalen Extrema „herumlaufen“ und folglich rechenzeitaufwendiges Steckenbleiben in lokalen Extrema vermeiden. Ähnlich funktioniert diese simple Erklärung in höheren Dimensionen: je höher die Dimension, desto mehr Umwege führen um lokale Extrema herum, und desto kleiner ist die Wahrscheinlichkeit in lokalen Extrema steckenzubleiben.

Wir vermuten, daß die vorgeschlagene Strategie vorteilhaft auf eine ganze Reihe von Optimierungsproblemen angewandt werden kann. Die wichtigste Eigenschaft dieser Aufgaben sollte sein, daß sie in andere Probleme transformiert werden kann, wobei die Lösung des erweiterten Problems mit einem Parameter  $N$  die Lösung  $N^*$  des ursprünglichen Problems als Spezialfall enthält. Unseres Erachtens gehört ein großer Teil der sogenannten Bedeckungsprobleme zu dieser Klasse. Dabei besteht die Aufgabe darin, ein gegebenes Gebiet mit einer kleinstmöglichen Zahl von kleineren Flächen vollständig zu bedecken. Die Kenntnis der Lösung einer solchen Bedeckungsaufgabe mit  $N$  Flächen ist sehr wahrscheinlich nützlich zur Lösung der Aufgabe mit  $N - 1$  Flächen.

problems” *Biol. Cyber.* **52**, 141–152.

*Lecture notes in computer science*, **840** (Springer, Berlin).

### 3.7 Robuste Mutationsverteilungen in kontinuierlichen Suchräumen

(C. Kappler)

Evolutionäre Algorithmen sollen *robust* sein. Das heißt, der Algorithmus soll zuverlässig ein beliebiges Problem optimieren – ohne daß erst geeignete Parametereinstellungen des Algorithmus gesucht werden müssen.

Wir wollen uns hier auf den Einfluß der Mutationsverteilung auf die Robustheit des Algorithmus konzentrieren. Im folgenden schlagen wir einige Eigenschaften von einfach zu ziehenden Mutationsverteilungen, die zu robusten Suchalgorithmen führen, vor (kurz: “robuste Mutationsverteilungen”). Wir untersuchen üblicherweise benutzte Mutationsverteilungen auf diese Eigenschaften hin, und konstruieren eine Familie von Verteilungen, die alle

genannten Eigenschaften aufweist. Schließlich testen wir alle Mutationsverteilungen am Rastriginproblem.

Eine im obigen Sinne annähernd robuste Mutationsverteilung wurde auch erfolgreich für eine Molekulardynamik Simulation eingesetzt (Details s. Kapitel 5.2).

Wir beschränken uns im folgenden auf *faktorisierende* Mutationsverteilungen  $P(\vec{x}) = \prod_i f(x_i)$ , d. h. Verteilungen, die erlauben, jede Koordinate unabhängig von den anderen zu ziehen. Dies macht das Ziehen bequemer.

Es ist intuitiv einleuchtend, daß robuste Mutationsverteilungen die folgenden Eigenschaften haben sollten:

- R. Keine bestimmte Richtung wird bevorzugt untersucht, d. h. die Verteilung muß *rotationssymmetrisch* sein. Auf diese Weise wird keine Annahme über die Geometrie der Suchlandschaft gemacht.
- G. Es ist möglich, gelegentlich weit entfernte Punkte des Suchraumes zu testen. Diese *globale Suchfähigkeit* verhindert, daß der Algorithmus in lokalen Optima steckenbleibt. Mathematisch ausgedrückt: die Mutationsverteilung muß für große Radien abfallen wie ein Potenzgesetz, und nicht exponentiell.
- L. Andererseits muß auch *lokale Suche* möglich sein, damit eine einmal gefundene Lösung verfeinert werden kann. Mathematisch ausgedrückt: die Mutationsverteilung muß sich für kleine Radien entwickeln wie  $P(r) \propto r^\alpha$ ,  $\alpha \leq 0$ .

All dies sind zunächst einmal Vorschläge. Weiter unten untersuchen wir, inwieweit jede dieser Eigenschaften zur Robustheit eines evolutionären Algorithmus beiträgt.

Zunächst betrachten wir die üblichste Art von Mutationen, bei der jede der  $n$  Koordinaten nach einer Gaußverteilung gezogen wird. Offenbar ist die Gesamtverteilung dann rotationssymmetrisch. Globale und lokale Suche sind hingegen nicht möglich: Die Wahrscheinlichkeit, einen bestimmten Radius zu ziehen ist  $\chi^2$  verteilt, und fällt demnach exponentiell ab, und steigt an wie  $r^{n-1}$  – d. h. sehr langsam. Gaußverteilte Mutationen sollten demnach nicht zu einem robusten Suchalgorithmus führen.

In der Literatur [SH87, YL96], unter anderem im Rahmen von EVOALG (s. [Kap96] und Zwischenberichte Nr. 3–5), wird vorgeschlagen  $n$ -dimensionale

Cauchyverteilung

$$P(\vec{x}) = \frac{1}{\pi^{\frac{n+1}{2}}} M \frac{\sigma}{(\sigma^2 + |\vec{x}|^2)^{\frac{n+1}{2}}}, \quad \sigma > 0, \quad M M \quad (3.52)$$

als Mutationsverteilung zu benutzen, Durch einen Trick ist es möglich, jede Koordinate getrennt zu ziehen. Auch diese Verteilung ist offenbar rotationssymmetrisch. Außerdem gilt

$$P(r) = \frac{2, \left(\frac{n+1}{2}\right)}{\left(\frac{n}{2}\right)\pi^{\frac{1}{2}}} r^{n-1} \frac{1}{(\sigma^2 + r^2)^{\frac{n+1}{2}}}, \quad (3.53)$$

d. h. die Gesamtverteilung fällt ab mit einem Potenzgesetz, und globale Suche ist möglich. Lokale Suche ist, aus den gleichen Gründen, wie bei der Gaußverteilung, jedoch nicht möglich.

Ingber [Ing89] schlug vor jede Koordinate nach

$$P(x_i) = \begin{cases} (2 \ln(\frac{x_i^{\text{cut}}}{\sigma} + 1))^{-1} \frac{1}{\sigma + |x_i|}, & |x_i| < x_i^{\text{cut}} \\ 0 & \text{otherwise} \end{cases} \quad (3.54)$$

zu ziehen. Diese Verteilung fällt ab wie  $1/x$  und kann daher nicht normalisiert werden, ohne an einem Wert  $x^{\text{cut}}$  abgeschnitten zu werden. Dadurch werden zusätzliche Parameter eingeführt. Die Gesamtverteilung ist in diesem Fall nicht rotationssymmetrisch. Globale Suche ist möglich, da  $P(r) \propto r^{-1}$ , wie weiter unten gezeigt wird. Lokale Suche scheitert aber, genau wie für die Gauß- und die Cauchyverteilung an einer Entwicklung  $P(r) \propto r^{n-1}$  für kleine  $r$ .

Um die bisherigen Ergebnisse zusammenzufassen: keine der bisher betrachteten Verteilungen hat alle von uns geforderten Eigenschaften, und sollten demnach nicht zu robusten Suchalgorithmen führen.

Wie sollte also eine Mutationsverteilung aussehen, die alle Eigenschaften (R.), (G.) und (L.) aufweist?

- R. um eine rotationssymmetrische Mutationsverteilung zu erhalten, darf die Gesamtverteilung nur eine Funktion des Radius sein:  $P(\vec{x}) = f(r)$ .

Man beachte, daß alle *faktorisierenden* rotationssymmetrischen Verteilungen die Form  $P(\vec{x}) \propto \exp[f(\sum x_i^2)]$ , solange man kartesische Koordinaten benutzt.

- G. Wenn jede Koordinate nach einer Verteilung gezogen wird, die abfällt wie  $P(r) \propto r^\alpha$ , dann gilt

$$P(r) \propto r^{n(1-\alpha)-1}, \quad r \text{ groß}, \quad (3.55)$$

Dieses Ergebnis erhält man über die Methode der Funktionen [vK81]. D. h. wenn die eindimensionale Verteilung, nach der die einzelnen Koordinaten gezogen werden, wie ein Potenzgesetz abfällt, dann fällt auch die Verteilung des Radius wie ein Potenzgesetz ab.

- L. Wenn jede Koordinate nach einer Verteilung gezogen wird, die für kleine  $x_i$  entwickelt wird wie  $P(x_i) = 1/2c + \text{dots}$ , mit  $c > 0$ , dann wird die Verteilung der Radien entwickelt wie

$$P(r) = \frac{1}{(2c)^n} S_n(r), \quad r \text{ klein}, \quad (3.56)$$

wobei  $S_n(r)$  die Oberfläche der  $n$ -dimensionalen Hyperkugel ist. Dieses Ergebnis kann durch direkte Integration und vollständige Induktion über  $n$  gezeigt werden. Es ist einfach zu verstehen: Es gibt genau einen Vektor  $\vec{x}$  vom Ursprung zu einem bestimmten Punkt auf der Oberfläche der Hyperkugel  $S_n(r)$ . Die Wahrscheinlichkeit, diesen Vektor zu ziehen ist  $1/(2c)^n$ , und es gibt  $S_n(r)$  solche Vektoren mit Radius  $r$ . Daher ist die Wahrscheinlichkeit, einen Vektor der Länge  $r$  zu ziehen wie oben angegeben. Das bedeutet, solange man *kartesische* Koordinaten einzeln nach einer nicht-singulären Verteilung zieht, bekommt man eine Gesamtverteilung die ansteigt wie  $P(r) \propto r^{n-1}$  oder langsamer – also keine lokale Suche.

Aus all dem folgt, daß man am besten  $n$ -dimensionale *sphärische* Koordinaten benutzt. Die Winkel zieht man gleichverteilt, und den Radius nach

$$P_\alpha(x) = \frac{\alpha\sigma^\alpha}{\pi} \sin\left(\frac{\pi}{\alpha}\right) \frac{1}{\sigma^\alpha + x^\alpha}, \quad \alpha > 1, \quad \sigma > 0, \quad (3.57)$$

Durch Variation von  $\alpha$  kann die Stärke des Abfalls, und damit die Intensität der globalen Suche variiert werden. Für  $\alpha = 2$  wird diese Verteilung zur Cauchyverteilung (s. Gl. 3.52), für  $\alpha = 1$  wird sie zu Ingber's Verteilung (s. Gl. 3.54).

Eine andere Möglichkeit Mutationsverteilungen mit den gewünschten Eigenschaften zu erhalten könnte in Schrittweitenadaptation [BS93, Rec73b] bestehen. Ein einfacher numerischer Test schien dies nahelegen. Detaillierte Untersuchungen zu diesem Punkt wurden allerdings nicht durchgeführt.

Bisher haben wir nur Plausibilitätsargumente präsentiert, um die geforderten Eigenschaften Rotationssymmetrie, globale, und lokale Suchfähigkeit zu rechtfertigen. In diesem Abschnitt stellen wir ein numerisches Experiment vor, in dem wir Mutationsverteilungen mit verschiedenen Eigenschaften testen.

Gesucht sind robuste Suchalgorithmen. In diesem Abschnitt wollen wir darunter die Fähigkeit verstehen, eine gute Lösung zu finden, ohne daß die Parameter des Algorithmus, insbesondere die Breite der Mutationsverteilung  $\sigma$ , speziell eingestellt werden müssen. Konkret messen wir die Zeit, die der Algorithmus mit Mutationsverteilung  $P(x)$  benötigt, eine Lösung mit Fitneß  $V > V_{stop}$  zu finden, wobei  $V_{stop}$  eine gute Lösung ist. Der robusteste Algorithmus hat (i) das breiteste Fenster in  $\sigma$ , in dem die Konvergenzzeit  $t(\sigma)$  kurz ist, und (ii) möglichst auch die kürzeste minimale Konvergenzzeit  $t_{min}$ . Da wir nur das Konvergenzverhalten in einer einzigen Landschaft betrachten, wird das in der Einleitung erwähnte Kriterium "robuste Konvergenz unabhängig von der Suchlandschaft" also ausgelassen. Stattdessen wählen wir ein als besonders schwierig bekanntes Testproblem.

Das Testproblem ist die Rastriginlandschaft mit der Fitneß

$$V(\vec{x}) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)] + 10n. \quad (3.58)$$

Wegen der großen Anzahl lokaler Maxima ist diese Landschaft als besonders schwierig bekannt.[Rec73b, Sch77a] Das globale Maximum ist im Ursprung mit  $V(\vec{0}) = 0$ . Der Einfachheit halber, und ohne Beschränkung der Allgemeinheit, benutzen wir eine (1+1) Strategie. Wir setzen die Dimension  $n = 10$  und starten den Algorithmus bei  $x_i = 10^2 \forall i$ , d. h.  $V = 10^5$ . Wir wählen  $V_{stop} = 500$ .

Die unten angegebenen Mutationsverteilungen wurden getestet, geordnet nach ihren Eigenschaften – dabei bedeutet "kartesisch", daß jede kartesische Koordinate  $x_i$  wird nach der angegebenen Verteilung gezogen wird. "Sphärisch" bedeutet, daß sphärische Koordinaten benutzt werden, mit gleichverteilten Winkeln, und einem Radius der nach der angegebenen Verteilung gezogen wird.

*keine* kartesischer Ingber (s. Gl. 3.54) mit  $x_i^{cut} = 10^{-1} \forall i$ , d. h. die Funktion wird, im Vergleich zur Gesamtdistanz, die zurückgelegt werden muß, extrem kurz abgeschnitten.

R. kartesischer Gauß.

G. kartesischer Cauchy (s. Gl. 3.52),  
kartesischer Ingber mit  $x_i^{cut} = 10^3 \forall i$ , d. h. die Verteilung wird bei einem

Wert abgeschnitten, der erheblich größer ist, als der größte der benötigt wird.

R,L. sphärischer Gauß

R,G,L. sphärischer Cauchy,

sphärischer Ingber mit  $r^{cut} = 10^3$ ,

sphärisches  $P_\alpha$  (s. Gl. 3.57) mit verschiedenen Werten für  $\alpha$ .

Abb. ?? zeigt das Ergebnis. Am besten schneiden die Verteilungen ab, die alle Kriterien erfüllen. Sie haben sowohl das breiteste Fenster in  $t(\sigma)$ , als auch die kleinsten Werte von  $t_{min}$ . Für die Verteilungen  $P_\alpha$  zeigen wir nur zwei repräsentative Beispiele. Für  $1 < \alpha < 2$  führen zu Kurven zwischen denen von sphärischen Cauchy ( $\alpha = 2$ ) und sphärischem Ingber ( $\alpha = 1$ ). Für  $\alpha > 2$  nähern sie sich der Kurve für sphärischen Gauß. Am schlechtesten schneidet der zu kurz abgeschnittene, kartesische Ingber, der keine der geforderten Eigenschaften besitzt, ab, gefolgt von kartesischem Gauß, der nur rotationssymmetrisch ist.

Nun betrachten wir den Einfluß der verschiedenen Eigenschaften im einzelnen. Die Unfähigkeit global zu suchen sollte sich bei kleinen Werten der Breite der Verteilung  $\sigma$  zeigen: Es sind dann relativ mehr Schritte nötig um zu konvergieren, weil der Algorithmus in lokalen Optima steckenbleibt. Tatsächlich hat die Kurve für sphärischen Gauß eine steilere (linke) Flanke, als die der anderen sphärischen Verteilungen. Zudem ist diese Flanke um so flacher, je langsamer die Mutationsverteilung abfällt, d. h. je mehr die globale Suche betont wird.

Dieses letzte Ergebnis würde man nicht von vorneherein erwarten. Die Frage ist: wie häufig sollte der Algorithmus global suchen? Je intensiver die globale Suche, desto mehr wird die lokale Suche vernachlässigt. Dieser Effekt wird (bei gleichem Abfallexponenten  $\alpha$ ) um so ausgeprägter, je höher die Dimension ist, da relativ immer mehr Volumen weiter weg vorhanden ist. Daher würde man eigentlich erwarten, daß es einen idealen Exponenten  $\alpha$  gibt, und Mutationsverteilungen, die langsamer oder schneller abfallen, weniger gute Ergebnisse liefern. Dies ist jedoch nicht der Fall. Im Bereich der getesteten Exponenten gilt: je langsamer der Abfall, desto besser, und damit schneidet Ingbers Funktion am besten ab.

Das Problem mit Ingbers Funktion ist jedoch die Wahl von  $x^{cut}$ . Wie aus Abb. ?? gut ersichtlich, verdirbt eine zu kleine Wahl dieses Wertes das Ergebnis. Eine gute Möglichkeit wäre eine Mutationsverteilung  $P_\alpha$  mit  $\alpha$  gerade größer eins, so daß die Funktion normierbar bleibt.

Die Unfähigkeit lokal zu suchen sollte sich hingegen bei großen Werten von  $\sigma$  zeigen: Je größer  $\sigma$ , desto größer der kleinste (noch mit einiger Wahr-

scheinlichkeit) mögliche Suchschritt. D. h. der Algorithmus kommt nur auf eine bestimmte Distanz an das Optimum heran. Tatsächlich, Kurven von Mutationsverteilungen, die lokale Suche ausschließen, d. h. die Kurven aller kartesischen Verteilungen, steigen für große  $\sigma$  erheblich steiler an, als die Kurven sphärischer Verteilungen.

Der Einfluß der Rotationssymmetrie ist aus diesem Experiment nicht ersichtlich. Man müßte eine Mutationsverteilung konstruieren, die sowohl lokal als auch global sucht, ohne rotationssymmetrisch zu sein.

Das Ziel dieser Arbeit war, robuste Evolutionäre Algorithmen zu konstruieren, d. h. Algorithmen, die zuverlässig für viele verschiedene Probleme funktionieren, ohne eine spezielle Einstellung ihrer Parameter zu benötigen. In diesem Zusammenhang haben wir die Rolle der Mutationsverteilungen betrachtet. Wir haben verschiedene Eigenschaften vorgeschlagen, die Mutationsverteilungen, die zu robusten Suchalgorithmen führen, erfüllen sollten: Rotationssymmetrie, globale Suchfähigkeit, und lokale Suchfähigkeit. Gemeinhin benutzte Mutationsverteilungen erfüllen nicht alle dieser Kriterien. Wir haben daher geeignete Verteilungen vorgeschlagen. Sie benutzen sphärische Koordinaten, und fallen für große Radien möglichst langsam ab, nach einem Potenzgesetz. Ein numerisches Experiment bestätigt, daß Mutationsverteilungen mit den vorgeschlagenen Eigenschaften zu den robustesten Suchalgorithmen führen.

### 3.8 Die analytische Formulierung der $(\mu, \lambda)$ -Strategie

### 3.9 Optimierung mittels Brownscher Teilchen

(B. Tilch)

Das Modell gepumpter Brownscher Teilchen wurde entwickelt, um Systeme zu modellieren, in denen die Energieaufnahme, -speicherung und -umsetzung eine relevante Modelleigenschaft ist [ESTCht]. Dies betrifft biologische Systeme, z. Bsp. Bakterien, aber technische Systeme wie autonome Roboter bzw. Fahrzeuge können auf einer abstrakten Ebene ebenfalls mit solchen Annahmen modelliert werden.

Da auch die Boltzmann Strategie zur Optimierung mit Brownschen Teilchen modelliert werden kann [AER97], stellt sich die Frage, ob auch das Modell Brownscher Teilchen mit innerem Energiespeicher in der Lage ist,

Optimierungsprobleme zu lösen und welche Gemeinsamkeiten und Unterschiede im Vergleich mit der Boltzmann Strategie vorhanden sind.

Der Energiespeicher  $e$  der Teilchen wird durch einen kontinuierlichen positiven Energiestrom  $q_0$  gefüllt. Mit einer konstanten Rate  $c$  dissipiert Energie aus dem Speicher. Weiterhin wird gespeicherte Energie mit einer Rate  $d(v)$  in kinetische Energie umgewandelt. Für  $d(v)$  wird der Ansatz

$$d(v) = d_2 v^2 \quad (3.59)$$

gewählt. Man erhält mit diesen Annahmen folgendes dynamisches Modell für ein Brownsches Teilchen, das am Ort  $x$  mit der Geschwindigkeit  $v$  und der Energiemenge  $e$  im Speicher angetroffen wird.

$$\begin{aligned} \dot{x} &= v \\ m\dot{v} &= -\gamma v - \nabla F + d_2 v e + \sqrt{2D}\xi(t) \\ \dot{e} &= q_0 - ce - d_2 v^2 e \end{aligned} \quad (3.60)$$

Der Parameter  $\gamma$  modelliert eine Reibungskraft,  $F$  ist die Fitneß, die gleichzeitig als Potential auf die Bewegung des Teilchen wirkt. Dieser Ansatz sucht also generell nach den Minima in der Fitneß.  $\xi(t)$  ist eine weiße Rauschquelle,  $\langle \xi(t) \rangle = 0$ ;  $\langle \xi(t), \xi(t') \rangle = \delta(t - t')$ , der Stärke  $D$ . Die Masse  $m$  wird im weiteren  $m = 1$  gesetzt.

Betrachtet man den Limes der Energieumsetzung gegen Null, d.h.  $d_2 \rightarrow 0$ , so erhält man das Modell einfacher Brownscher Teilchen in einem Kraftfeld  $F$ , das im überdämpften Fall der Boltzmann Strategie entspricht.

Es wurde numerisch der Fall eines Doppeltopfpotentials mit der Gleichung  $F(x) = 0.16 + 0.0278x^2(0.16667x - 0.9)(0.16667x + 0.8)$  untersucht. Bei diesem Potential liegt das globale Minimum auf der positiven Halbachse und ein lokales auf der negativen. Es wurden 10000 Teilchen simuliert, die alle mit leerem Energiespeicher am Ort  $x = 0$  mit  $(0, D/\gamma)$ -gaußverteilten Geschwindigkeiten starteten. Es wurden Ortsverteilungen und die mittlere Fitneß für verschiedene Werte des Energieübertrages  $d_2$  untersucht. Durch die obige Wahl der Parameter sind die Ergebnisse der Simulationen mit denen aus [AER97] vergleichbar. In Abb. 3.18 ist die zeitliche Entwicklung der mittlerem Fitneß aufgetragen.

Für den Fall verschwindenden Energieübertrages gleicht die zeitliche Entwicklung von  $\langle F \rangle$  der einer Boltzmannstrategie. Erhöht man den Parameter  $d_2$  fällt die mittlere Fitneß zu Beginn der Simulation schneller ab, erreicht aber dann im Grenzwert großer Zeiten einen schlechteren Wert. Weiterhin erkennt man, das bei höheren Werten für  $d_2$  ein Einschwingvorgang durchlaufen wird, dessen Amplitude ebenfalls mit  $d_2$  ansteigt. Bei allen Simulationen finden die Teilchen das globale Minimum.

In Abb. 3.19 ist die Ortsverteilung bei verschiedenen Zeiten für zwei Werte von  $d_2$  aufgetragen. Für den Fall  $d_2 = 0.0$  ergibt sich die gleiche Entwicklung der Ortsverteilung wie bei einer Boltzmann-Strategie mit kleiner Temperatur. Für nicht verschwindenden Energieübertrag verschmelzen die beiden Maxima der Verteilung und es ergibt sich eine breite Verteilung. Qualitativ gleichartige eingipflige Verteilungen ergeben sich bei Durchführung einer Boltzmann-Strategie mit hoher Temperatur.

Die Annahme, daß die Sucher einen inneren Energiespeicher besitzen, aus dem kontinuierlich Energie in kinetische Energie umgewandelt wird, erzeugt eine dem Erhöhen der Temperatur äquivalente Ortsverteilung der Sucher.

Eine breitere Verteilung bedeutet, daß die Teilchen lokale Minima schneller verlassen können, da sie mit größerer Wahrscheinlichkeit, die sie umgebenden Potentialberge hinaufwandern können. Die Hinzunahme des Energiespeichers kann also Boltzmann-Suchern das Verlassen lokaler Minima auch bei tiefen Temperaturen ermöglichen.

Im vorigen Abschnitt interessierte vor allem der Vergleich des Modells mit der Boltzmann Strategie. Aufgrund der Anfangsbedingungen fanden die Teilchen immer das globale Minimum. In diesem Abschnitt soll untersucht werden, ob das globale Minimum auch bei ungünstigen Anfangsbedingungen gefunden wird. Dazu wurde die als Startposition aller Teilchen  $x = -5$  gewählt, so daß die Teilchen zuerst das lokale Minimum bei  $x = -3.8$  finden. Anhand der zeitlichen Entwicklung der kleinsten Fitneßwerte in Abb. 3.20 sieht man, daß mit steigendem Energieübertrag  $d_2$  das lokale Minimum schneller verlassen wird und somit das globale Minimum in kürzerer Zeit gefunden wird.

Es wurde gezeigt, daß mit Hilfe des Modells Brownscher Teilchen mit innerem Energiespeicher Optimierungsprobleme prinzipiell gelöst werden können. Dieses Modell hat als Grenzfall verschwindenden Energieübertrages vom Speicher in kinetische Energie die bekannte Boltzmann Strategie. Das Hinzunehmen eines Energieübertrages bei festgehaltenen Parametern entspricht entspricht einer Temperaturerhöhung im Fall der Boltzmann Strategie.

Es wurde am einfachen Beispiel eines Doppelpotentials gezeigt, das das globale Optimum mit steigendem Energieübertrag schneller gefunden wird. Die Ortsverteilung wird breiter und die mittlere Fitneß nimmt zu. Deshalb können die Teilchen leichter das lokale Minimum verlassen.

## 3.10 Erstellte Software

### 3.10.1 PDE2D

(U. Erdmann)

Im Verlaufe des Projekts wurde, basierend auf der FORTRAN Routine PDETWO, ein Programm zur Simulation von Systemen partieller Differentialgleichungen parabolischen Typs für räumlich 2-dimensionale Probleme mit globaler Kopplung erstellt, die sich wie folgt formulieren lassen:

$$\frac{\partial u}{\partial t} = f \left( t, x, y, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial y^2} \right) + \int F(x, y, t) u(x, y, t) dx dy$$

Die einzelnen Variablen bedeuten:

$$\begin{aligned} u &= \left( u_1 \quad \cdots \quad u_n \right) = f(t, x, y) \\ \frac{\partial u}{\partial t} &= \left( \left( \frac{\partial u}{\partial t} \right)_1 \quad \cdots \quad \left( \frac{\partial u}{\partial t} \right)_n \right) \\ \frac{\partial u}{\partial x} &= \left( \left( \frac{\partial u}{\partial x} \right)_1 \quad \cdots \quad \left( \frac{\partial u}{\partial x} \right)_n \right) \\ \frac{\partial u}{\partial y} &= \left( \left( \frac{\partial u}{\partial y} \right)_1 \quad \cdots \quad \left( \frac{\partial u}{\partial y} \right)_n \right) \\ \frac{\partial^2 u}{\partial x^2} &= \begin{pmatrix} \left( \frac{\partial^2 u}{\partial x^2} \right)_{11} & \cdots & \left( \frac{\partial^2 u}{\partial x^2} \right)_{1n} \\ \vdots & \ddots & \vdots \\ \left( \frac{\partial^2 u}{\partial x^2} \right)_{n1} & \cdots & \left( \frac{\partial^2 u}{\partial x^2} \right)_{nn} \end{pmatrix} = \begin{pmatrix} \frac{\partial}{\partial x} \left( D_{11}^h \left( \frac{\partial u}{\partial x} \right)_1 \right) & \cdots & \frac{\partial}{\partial x} \left( D_{1n}^h \left( \frac{\partial u}{\partial x} \right)_1 \right) \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial x} \left( D_{n1}^h \left( \frac{\partial u}{\partial x} \right)_n \right) & \cdots & \frac{\partial}{\partial x} \left( D_{nn}^h \left( \frac{\partial u}{\partial x} \right)_n \right) \end{pmatrix} \\ \frac{\partial^2 u}{\partial y^2} &= \begin{pmatrix} \left( \frac{\partial^2 u}{\partial y^2} \right)_{11} & \cdots & \left( \frac{\partial^2 u}{\partial y^2} \right)_{1n} \\ \vdots & \ddots & \vdots \\ \left( \frac{\partial^2 u}{\partial y^2} \right)_{n1} & \cdots & \left( \frac{\partial^2 u}{\partial y^2} \right)_{nn} \end{pmatrix} = \begin{pmatrix} \frac{\partial}{\partial y} \left( D_{11}^v \left( \frac{\partial u}{\partial y} \right)_1 \right) & \cdots & \frac{\partial}{\partial y} \left( D_{1n}^v \left( \frac{\partial u}{\partial y} \right)_1 \right) \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial y} \left( D_{n1}^v \left( \frac{\partial u}{\partial y} \right)_n \right) & \cdots & \frac{\partial}{\partial y} \left( D_{nn}^v \left( \frac{\partial u}{\partial y} \right)_n \right) \end{pmatrix} \end{aligned}$$

Es sind Probleme mit Kreuzdiffusion lösbar:

$$\begin{aligned} D^h &= \begin{pmatrix} D_{11}^h & \cdots & D_{1n}^h \\ \vdots & \ddots & \vdots \\ D_{n1}^h & \cdots & D_{nn}^h \end{pmatrix} = f(t, x, y, u) \\ D^v &= \begin{pmatrix} D_{11}^v & \cdots & D_{1n}^v \\ \vdots & \ddots & \vdots \\ D_{n1}^v & \cdots & D_{nn}^v \end{pmatrix} = f(t, x, y, u) \end{aligned}$$

Die Randbedingungen können wie folgt angegeben werden:

$$\begin{aligned} a^h u + b^h \frac{\partial u}{\partial y} &= c^h \\ a^v u + b^v \frac{\partial u}{\partial x} &= c^v \end{aligned}$$

$a, b, c$  sind Vektoren der Länge  $n$ , wie  $u$  und  $\frac{\partial u}{\partial x}$ , deren Komponenten wieder von  $t, x, y$  abhängen. Das Programm verwendet die Routine PDETWO aus dem Paket PDELIB (NIST Guide to Available Math Software), die als freie Software erhältlich ist.

Im Gegensatz zu meist verwendeten schnellen expliziten Lösungsverfahren verwendet diese Routine auch implizite Diskretisierungsmethoden. Der Vorteil der impliziten Verfahren ist die Stabilität der numerischen Lösung unabhängig von der Wahl des Zeitschrittes. Während bei expliziten Verfahren deren Stabilität immer vom Verhältnis Zeit- zu Ortsdiskretisierung, z.B. im Falle des Fünfpunkt-LAPLACE-Operators:

$$\frac{\Delta t}{(\Delta r)^2} \leq \frac{1}{4} \quad (3.61)$$

abhängt, d.h. bei höherer Auflösung im Ortsraum muß ich den Zeitschritt entsprechend stärker verringern, so daß die Rechenzeit sich um einiges verzögert, ist das implizite Verfahren für alle Quotienten  $\frac{\Delta t}{(\Delta r)^2}$  stabil, was die freie Wahl der Ortsgenauigkeit läßt. Will man ein Problem auf großen Zeitskalen mit hoher Genauigkeit lösen, sind implizite Verfahren aufgrund ihrer Stabilität beim Lösungsverhalten im Vorteil. Liegt der Quotient  $\frac{\Delta t}{(\Delta r)^2}$  jedoch im Bereich der Stabilität des expliziten Verfahrens, ist dieses schneller als das implizite Verfahren mit demselben Quotienten, da beim impliziten Verfahren bei jedem Zeitschritt ein  $N$ -faches ( $N$  ist die Anzahl der Gitterpunkte im Ortsraum) simultanes Gleichungssystem, welches auch mehr Arbeitsspeicherplatz zum Abspeichern der Matrizen benötigt, gelöst wird.

Das erstellte Programm läßt dem Anwender die Wahl, ob ein explizites (hier Rückwärtsdifferenzen) oder implizites Verfahren (hier ADAMS-Methode) verwendet werden soll. Oben erklärte Vorteile der einzelnen Verfahren kann der Anwender also frei abwägen.

Zur weiteren Optimierung der Rechenzeit ist in PDE2D adaptive Schrittweitenanpassung bezüglich des Zeitschrittes integriert. Die partielle Differentialgleichung wird dabei wie folgt betrachtet:

$$A_t = f(x, y, A_x, A_y, A_{xx}, A_{yy}, A_{xy})$$

mit  $A = A(x, y, t)$  und  $A_{xy} = \frac{\partial^2 A}{\partial x \partial y}$ . In dieser Form kann man  $A_t$  als gewöhnliche Differentialgleichung betrachten, wenn die abhängigen Variablen schon bestimmt sind. Die adaptive Schrittweitenanpassung bezieht sich jetzt auf die Diskretisierung dieser gewöhnlichen Differentialgleichung. Wenn  $f(\dots)$  stark gekrümmt ist, wird der Zeitschritt runtergesetzt, während bei geringer Krümmung von  $f(\dots)$  ein großer Zeitschritt gewählt wird.

Wie schon erwähnt ist PDE2D in der Lage Systeme mit globaler Kopplung zu lösen. Zur Diskretisierung des Integrals wurde die erweiterte SIMPSON-Regel verwendet. (siehe "Numerical Recipes Fortran")

Zur erleichterten Handhabung wurde noch ein shell script erstellt, welches die einzelnen kommentierten Programmteile in einen Editor lädt und letztendlich PDE2D mit dem konkreten Problem implementiert startet.

### 3.10.2 ArtEvo

**NAME:** libea.a

#### **BESCHREIBUNG:**

*ArtEvo* ist eine Programmbibliothek zur Implementierung beliebiger Strategien auf beliebigen Suchräumen. Folgende Suchräume sind eingebunden:

- N-dimensionaler reeller Raum
- N-dimensionaler ganzzahliger Raum
- Sequenzräume über beliebigen Alphabet
- Raum aller Graphen über N Punkten

#### **Fitneßfunktionen:**

engel Engel-Sequenzen  
 merit LABS-Problem  
 Netzwerkproblem  
 parabel N-dimensionale Parabel  
 hill N-dimensionale umgekehrte Parabel  
 rosenbrock 2-dimensionale Rosenbockfunktion  
 treppe N-dimensionale Treppenfunktion  
 shekel 2-dimensionale Shekelfunktion  
 rastrigin N-dimensionale Rastriginfunktion  
 sparabel N-dimensionale Schwefels Parabel

schwefel N-dimensionale Schwweifunktion  
griewangk N-dimensionale Griewangkfunktion  
gauss N-dimensionale Gaußsche Zufallslandschaft  
dwell N-dimensionaler Doppelpf

Die einzelnen Optimierungsstrategien sind in den folgenden Programmen realisiert.

**AUTOREN:** Maik Kschischo, Helge Rosé (rose@summa.physik.hu-berlin.de)

### 3.10.3 Mix

**NAME:** mix [-h][-d ;dir<sub>i</sub>][-v ;level<sub>i</sub>]

**OPTIONEN:**

-h : Hilfe  
-d : Arbeitsdirectory mit Parameterfile mix.para  
-v : Verboseflag

**BESCHREIBUNG:**

Das Programm implementiert Algorithmen zur Optimierung von Problemen mit N-dimensionalem reellen Suchraum. Folgende Strategien sind verfügbar:

- *anneal*: Boltzmannstrategie mit und ohne Annealing.
- *mixed*: Gemischte Strategie mit und ohne Annealing.
- *adaptive*: Self Annealing Strategie.
- *turnier*: Darwinstrategie mit Turniersélection.
- *perfmix*: Performancebestimmung der gemischten Strategie für verschiedene Populationsgrößen und Laufzeiten.

**PARAMETER:** Parameterfile: mix.para

Runs Anzahl der Runs  
Populationsize Populationsgröße  
Dimension Problemdimension

Generations Generationen  
 x0 Startpunkt  
 dx0 Mutationsschrittweite  
 Delta Anfangsvarianz der Population  
 Temperature Temperatur  
 Tvariance Temperaturvarianz  
 Annealing Annealingfaktor  
 TMutStrength Mutationsstärke der Temperatur  
 TMutProb Mutationswahrscheinlichkeit der Temperatur  
 TselProb Selektionswahrscheinlichkeit der Temperatur  
 SelProb Selektionswahrscheinlichkeit  
 MutProb Mutationswahrscheinlichkeit  
 ABfile File der Anfangsbedingung  
 Writestep Speicherperiode  
 Savemode Speichermode  
 Mode Optimierungsstrategie  
 Fitneß Fitneßfunktion

**AUTOR:** Helge Rosé (rose@summa.physik.hu-berlin.de)

### 3.10.4 XMix

**NAME:** xmix [-h][-d |dir<sub>i</sub>][-v |level<sub>i</sub>]

#### OPTIONEN:

-h : Hilfe  
 -d : Arbeitsdirectory mit Parameterfile mix.para  
 -v : Verboseflag

#### BESCHREIBUNG:

Graphische Version von *mix* auf der Basis der Graphikbibliothek *XForms*. Als Strategien sind verfügbar:

- *anneal*: Boltzmannstrategie mit und ohne Annealing.
- *mixed*: Gemischte Strategie mit und ohne Annealing.
- *adaptive*: Self Annealing Strategie.
- *turnier*: Darwinstrategie mit Turniersélection.

- *perfmix*: Performancebestimmung der gemischten Strategie für verschiedene Populationsgrößen und Laufzeiten.

**PARAMETER:** Parameterfile: mix.para

Runs Anzahl der Runs  
Populationsize Populationsgröße  
Dimension Problemdimension  
Generations Generationen  
x0 Startpunkt  
dx0 Mutationsschrittweite  
Delta Anfangsvarianz der Population  
Temperature Temperatur  
Tvariance Temperaturvarianz  
Annealing Annealingfaktor  
TMutStrength Mutationsstärke der Temperatur  
TMutProb Mutationswahrscheinlichkeit der Temperatur  
TSelProb Selektionswahrscheinlichkeit der Temperatur  
SelProb Selektionswahrscheinlichkeit  
MutProb Mutationswahrscheinlichkeit  
ABfile File der Anfangsbedingung  
Writestep Speicherperiode  
Savemode Speichermode  
Mode Optimierungsstrategie  
Fitneß Fitnessfunktion

**AUTOR:** Helge Rosé (rose@summa.physik.hu-berlin.de)

### 3.10.5 SAS

**NAME:** sas [-h][-d ;dir;][-v ;level;]

**OPTIONEN:**

-h : Hilfe  
-d : Arbeitsdirectory mit Parameterfile mix.para  
-v : Verboseflag

**BESCHREIBUNG:**

Graphische verbesserte Version von *mix* auf der Basis der Graphikbibliothek *XForms*. Folgende Strategien sind verfügbar:

- *anneal*: Boltzmannstrategie mit und ohne Annealing.
- *mixed*: Gemischte Strategie mit und ohne Annealing.
- *adaptive*: Self Annealing Strategie.
- *turnier*: Darwinstrategie mit Turniersélection.
- *perfmix*: Performancebestimmung der gemischten Strategie für verschiedene Populationsgrößen und Laufzeiten.

**PARAMETER:** Parameterfile: mix.para

Runs Anzahl der Runs  
 Populationsize Populationsgröße  
 Dimension Problemdimension  
 Generations Generationen  
 x0 Startpunkt  
 dx0 Mutationsschrittweite  
 Delta Anfangsvarianz der Population  
 Temperature Temperatur  
 Tvariance Temperaturvarianz  
 Annealing Annealingfaktor  
 TMutStrength Mutationsstärke der Temperatur  
 TMutProb Mutationswahrscheinlichkeit der Temperatur  
 TSelProb Selektionswahrscheinlichkeit der Temperatur  
 SelProb Selektionswahrscheinlichkeit  
 MutProb Mutationswahrscheinlichkeit  
 ABfile File der Anfangsbedingung  
 Writestep Speicherperiode  
 Savemode Speichermode  
 Mode Optimierungsstrategie  
 Fitness Fitneßfunktion

**AUTOR:** Helge Rosé (rose@summa.physik.hu-berlin.de)

### 3.10.6 MMix

**NAME:** mmix [-h][-d |dir<sub>i</sub>][-v |level<sub>i</sub>]

**OPTIONEN:**

## 90KAPITEL 3. OPTIMALE ALGORITHMEN UND ADAPTIVE STRATEGIEN

-h : Hilfe  
-d : Arbeitsdirectory mit Parameterfile mix.para  
-v : Verboseflag

### **BESCHREIBUNG:**

Das Programm optimiert das LABS-Problem mit Hilfe der gemischten Strategie. Folgende Strategien sind verfügbar:

- *anneal*: Boltzmannstrategie mit und ohne Annealing.
- *mixed*: Gemischte Strategie mit und ohne Annealing.
- *adaptive*: Bestimmung der Zustandsdichte mit der Boltzmannstrategie.

**PARAMETER:** Parameterfile: mix.para

Populationsize Populationsgröße  
Dimension Problemdimension  
Generations Generationen  
Temperature Temperatur  
Tvariance Temperaturvarianz  
Selection Selektionswahrscheinlichkeit  
Mutation Mutationswahrscheinlichkeit  
abfile File der Anfangsbedingung  
Writestep Speicherperiode  
Savemode Speichermode  
Mode Optimierungsstrategie  
Fitness Fitneßfunktion  
Runs Anzahl der Runs

**AUTOR:** Helge Rosé (rose@summa.physik.hu-berlin.de)

### **3.10.7 XBola**

**NAME:** xbola [-hdmprstexv]

**OPTIONEN und PARAMETER:**

- h : Hilfe
- l : laufende Ausgabe
- n  $\langle \text{dim} \rangle$  : Problemdimension
- m  $\langle \text{max} \rangle$  : maximale Fitneßdifferenz
- s  $\langle \text{size} \rangle$  : Populationsgröße
- r  $\langle \text{runs} \rangle$  : Anzahl der Runs
- e  $\langle \text{time} \rangle$  : Zeitschritt
- g  $\langle \text{space} \rangle$  : Ortsschrittweite
- w  $\langle \text{wtime} \rangle$  : Speicherperiode
- t  $\langle \text{time} \rangle$  : Laufzeit
- d  $\langle \text{diff} \rangle$  : Diffusionskonstante
- b  $\langle \text{beta} \rangle$  : inverse Temperatur
- x  $\langle x(0) \rangle$  : Startpunkt
- v  $\langle \text{var} \rangle$  : Varianz
- f  $\langle \text{name} \rangle$  : Fitneßfunktion

**BESCHREIBUNG:**

Das Programm implementiert die Boltzmannstrategie als Langevinalgorithmus [AER97]. Damit ist der Vergleich zwischen den erzielten theoretischen Resultaten und dem Verhalten des Optimierungsalgorithmus möglich.

**AUTOR:** Helge Rosé (rose@summa.physik.hu-berlin.de)

**3.10.8 Fema**

**NAME:** fema [-hdnmprstexv]

**OPTIONEN und PARAMETER:**

-h : Hilfe  
 -l : laufende Ausgabe  
 -n  $\{dim\}_i$  : Problemdimension  
 -p  $\{pop\}_i$  : Anzahl der Populationen  
 -m  $\{max\}_i$  : maximale Fitneßdifferenz  
 -s  $\{size\}_i$  : Populationsgröße  
 -r  $\{runs\}_i$  : Anzahl der Runs  
 -w  $\{wtime\}_i$  : Speicherperiode  
 -t  $\{time\}_i$  : Laufzeit  
 -d  $\{diff\}_i$  : Diffusionskonstante  
 -x  $\{x(0)\}_i$  : Startpunkt  
 -v  $\{var\}_i$  : Varianz

### BESCHREIBUNG:

Das Programm implementiert die Darwinstrategie als Algorithmus auf Basis der Mastergleichung [AER97]. Damit ist der Vergleich zwischen den erzielten theoretischen Resultaten und dem Verhalten des Optimierungsalgorithmus möglich.

**AUTOR:** Helge Rosé (rose@summa.physik.hu-berlin.de)

### 3.10.9 Xfema

**NAME:** xfema [-hdnmpstexv]

### OPTIONEN und PARAMETER:

-h : Hilfe  
 -l : laufende Ausgabe  
 -n  $\{dim\}_i$  : Problemdimension  
 -p  $\{pop\}_i$  : Anzahl der Populationen  
 -m  $\{max\}_i$  : maximale Fitneßdifferenz  
 -s  $\{size\}_i$  : Populationsgröße  
 -r  $\{runs\}_i$  : Anzahl der Runs  
 -w  $\{wtime\}_i$  : Speicherperiode  
 -t  $\{time\}_i$  : Laufzeit  
 -d  $\{diff\}_i$  : Diffusionskonstante  
 -x  $\{x(0)\}_i$  : Startpunkt  
 -v  $\{var\}_i$  : Varianz

**BESCHREIBUNG:**

Graphische Version von *fema* auf der Basis der Graphikbibliothek *XForms*.

**AUTOR:** Helge Rosé (rose@summa.physik.hu-berlin.de)

**3.10.10 ProEvo**

**NAME:** proevo [-h][-d ;dir;][-f ;file;][-v ;level;]

**OPTIONEN:**

-h : Hilfe  
 -d : Arbeitsdirectory mit Parameterfile  
 -f : Parameterfile  
 -v : Verboseflag

**BESCHREIBUNG:**

Das Programm integriert die partiellen Differentialgleichungen aller untersuchten Strategien [AER97]:

- *BDS*: Boltzmannstrategie, Darwinstrategie und gemischte Strategie
- *ADS*: Darwinstrategie mit Schrittweitensteuerung
- *SAS*: Self Annealing Strategie

**PARAMETER:** Parameterfile: mix.para

Mode Strategie  
 Fitness Fitneßfunktion  
 Initfile Anfangsverteilung  
 Boxes Boxen  
 Steps Rechenschritte  
 Diffusion\_x Diffusion auf dem Suchraum  
 Diffusion\_y Diffusion des Steuerparameters  
 Gamma Selektionsrate  
 Kappa Selektionsrate für SAS  
 X0 Startpunkt  
 Xa Intervallanfang

Xb Intervallende  
Xsig Startvarianz  
X0 Startpunkt des Steuerparameters  
Xa Intervallanfang des Steuerparameters  
Xb Intervallende des Steuerparameters  
Xsig Startvarianz des Steuerparameters  
Wsteps Speicherperiode  
Savemode Speichermode  
Minhuge minimaler HUE-Farbparameter  
Maxhuge maximaler HUE-Farbparameter  
Saturation Farbsättigung  
Value Helligkeit

**AUTOR:** Helge Rosé (rose@summa.physik.hu-berlin.de)

### 3.10.11 Das Programm evo

**Autor** Friedemann Schautz (schautz@summa.physik.hu-berlin.de)

**Zweck** Optimierung auf kontinuierlichen Räumen beliebiger Dimension mit  $(\mu, \lambda)$  oder  $(\mu + \lambda)$  Strategie (nach Schwefel).

**Plattform** C++/UNIX (erfordert gmake, g++ empfohlen)

**Installation** Nach auspacken des tar-files gibt es 3 Verzeichnisse:

- include header-files sofern nicht in src
- lib ein paar Hilfsfunktionen
- src Quelltext

Compilieren:

```
cd lib
gmake
cd ../src
gmake evo # Version ohne Debugging information
# oder
gmake evo.g # Version mit Debugging information
```

**Ausführen** Hilfe:

`evo -h` bringt kurze Hilfe (Optionen und mögliche Eingabeparameter)

normaler run:

`evo -f <parameter-file> -o <output-file>`

debugging:

`evo.g -f <parameter-file> -o <output-file> -d <debugging level>`

`> <debugging-output>`

(je höher `<debugging level>` desto mehr Ausgaben werden erzeugt)

**Parameter-file** enthält Zeilen der Form

`Variablenname = Wert # Kommentar`

wird eine Variable nicht angegeben, erhält sie einen Standardwert (siehe `evo -h`).

Eine Erklärung aller Variablen, die im Parameter-file vorkommen können, befindet sich im `readme-file`.

**Ausgabeformat** Die Ausgabe erfolgt in das mit `-o` angegebenen file oder nach standard output. Sie enthält die Werte aller Variablen und die Ausgaben aller runs (siehe `pmode`, `runs`).

```
# Variable = Wert
...
# Variable = Wert
Generation  Fitness [ Koordinaten] ...   (je nach <pmode>)
...
# END ( Nr des runs )
<Leerzeile>
Generation  Fitness [ Koordinaten] ...
...
```

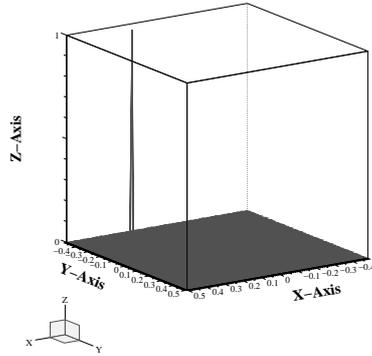
**Wahl der Fitness-Landschaft** Eine Fitness-Landschaft wird durch eine von `d_vector_species` abgeleitete Klasse repräsentiert. Beispiele befinden sich in den files `tests.h` und `more_tests.h`. Die Fitness-Landschaft wird zur compile-Zeit festgelegt. Dazu muß das Makro `TSPEC` den Namen der entsprechenden Klasse enthalten. Die Definition dieses Makros kann im file `landscape.h` oder als compiler-option (`-DTSPEC=name`) erfolgen.

Die Definition einer neuen Fitness-Landschaft ist am Anfang des files `tests.h` im Verzeichnis `src` beschrieben.

# 96 KAPITEL 3. OPTIMALE ALGORITHMEN UND ADAPTIVE STRATEGIEN

Wed Oct 11 17:48:33 1995

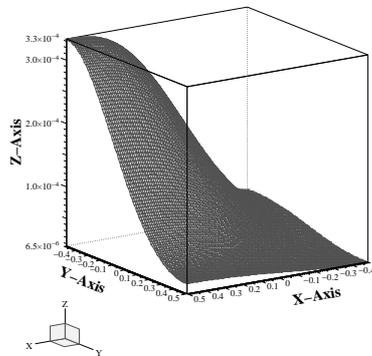
PLOT



Anfangsverteilung

Wed Oct 11 17:54:45 1995

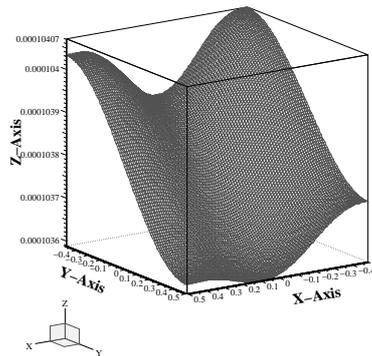
PLOT



Zeitschritt 8

Wed Oct 11 17:51:55 1995

PLOT



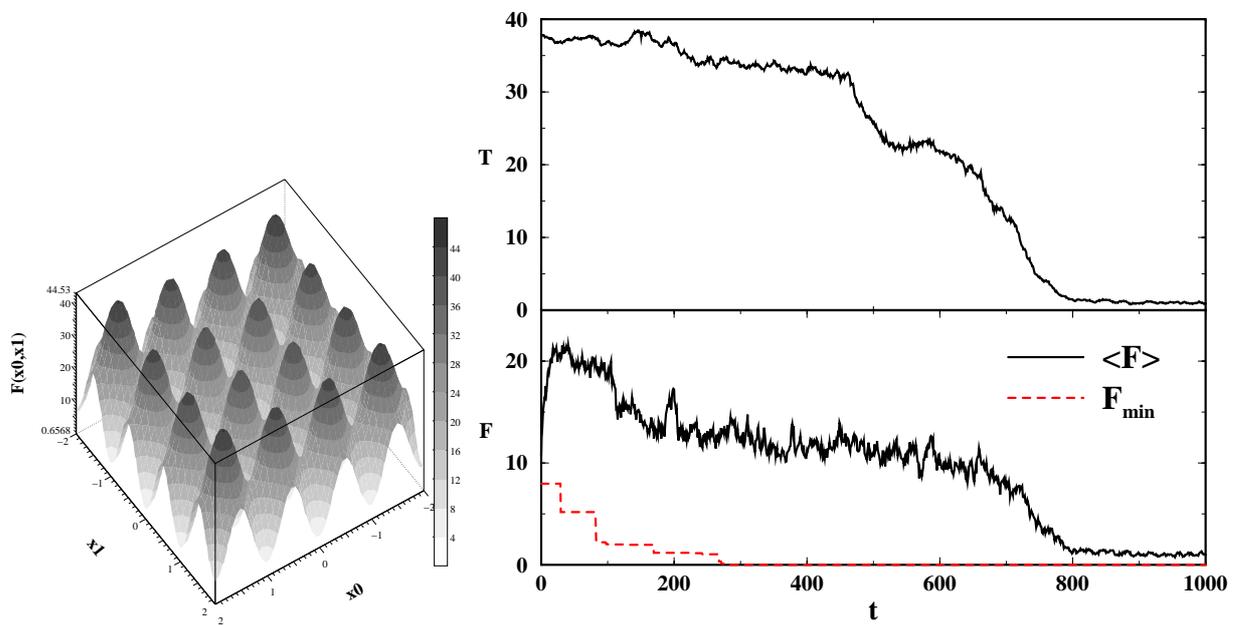


Abbildung 3.9: Minimierung der 2d-Rastriginfunktion (links) durch den Self Annealing Algorithmus mit 50 Suchern. Rechts: Die Entwicklung der mittleren Fitneß  $\langle F \rangle$ , mittleren Temperatur  $T$  und des Besten  $F_{min}$  für 1000 Optimierungsschritte.

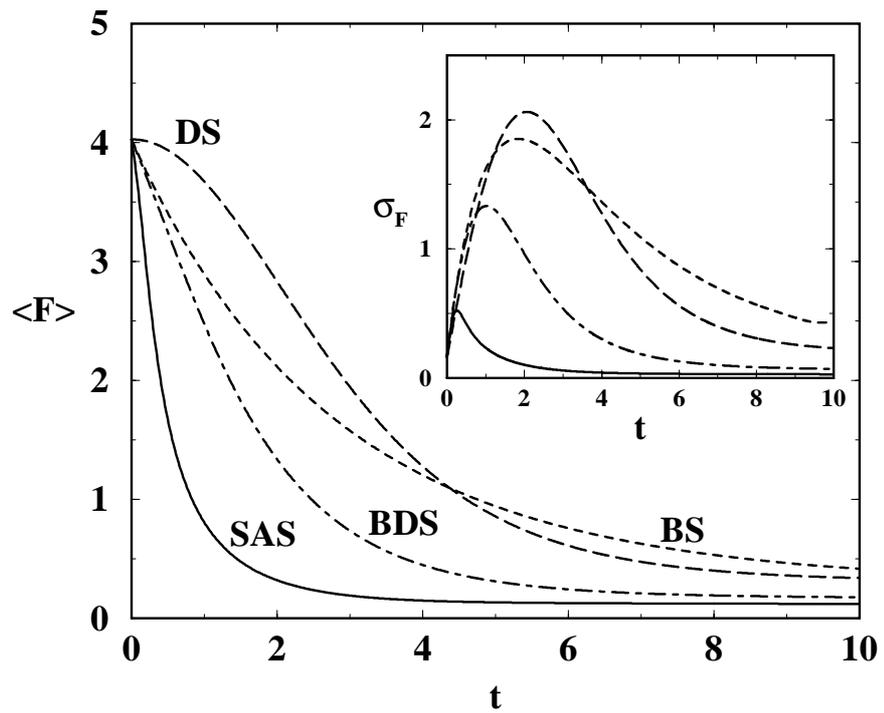


Abbildung 3.10: Entwicklung der mittleren Fitness und Standardabweichung der Boltzmann (BS), Darwin (DS), gemischte (BDS) und der Self Annealing Strategie (SAS) für die Minimierung von  $F(x) = x^2$ . Numerisch gelöst für:  $\beta = 1$ ,  $\gamma = 0,5$ ,  $\kappa = 3$ ,  $D = 0,1$ ,  $D_\beta = 0,01$ .

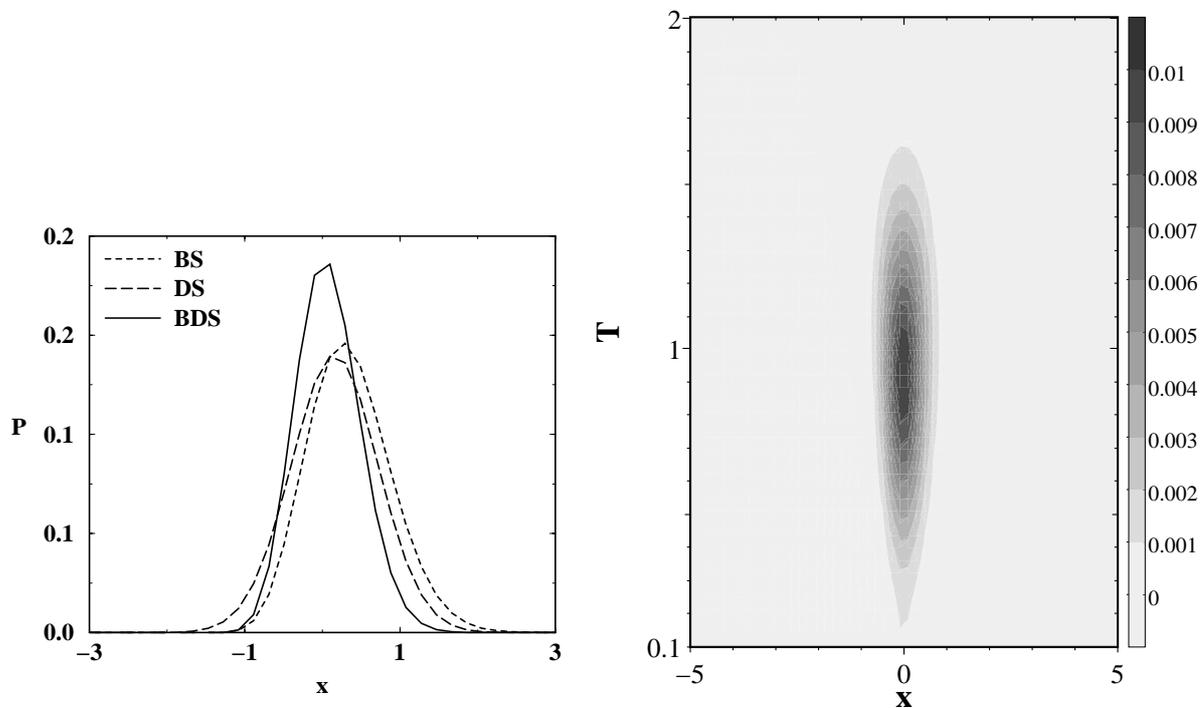


Abbildung 3.11: Wahrscheinlichkeitsverteilungen der Strategien nach  $t = 10$ . Links:  $P(x)$  für Boltzmann (BS), Darwin (DS), gemischte Strategie (BDS). Rechts:  $P(x, T)$  der Self Annealing Strategie.

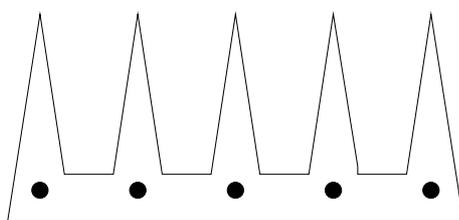


Abbildung 3.12: Der Grundriß eines Raumes mit 15 Wänden, zu dessen Beleuchtung  $N = \lfloor 15/3 \rfloor = 5$  Lampen benötigt werden. Die Punkte zeigen die Positionen der Lampen.

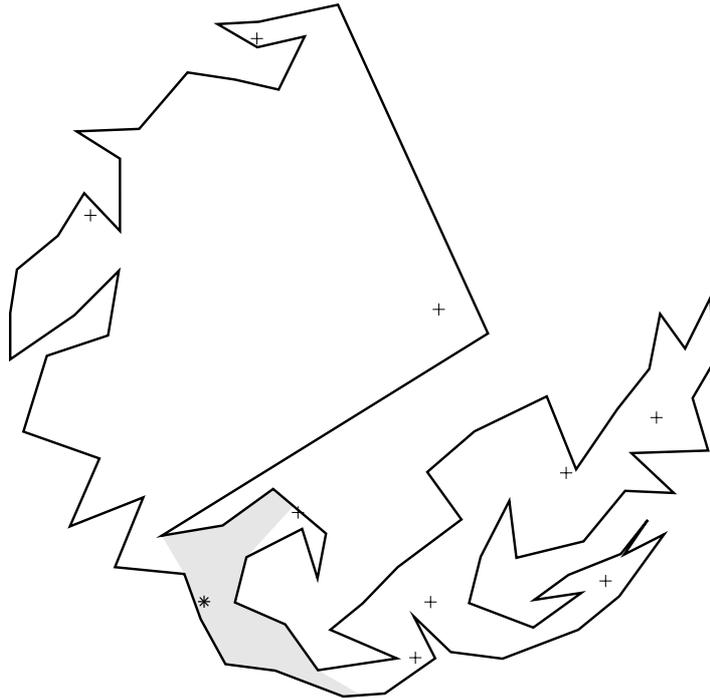


Abbildung 3.13: Lösung mit  $N_{min} = 10$  Lampen zur Beleuchtung eines Raumes mit  $M = 82$  Wänden. Die Positionen der Lampen sind mit (+)- und (\*)-Symbolen bezeichnet. Der Raum ist vollständig beleuchtet – man kann schnell prüfen, daß jeder Punkt der Wand mit mindestens einer der Lampen durch eine gerade Linie verbunden werden kann, ohne eine der Wandlinien zu schneiden. Die schattierte Fläche zeigt die Fläche, die durch die mit (\*) gekennzeichnete Lampe beleuchtet wird.

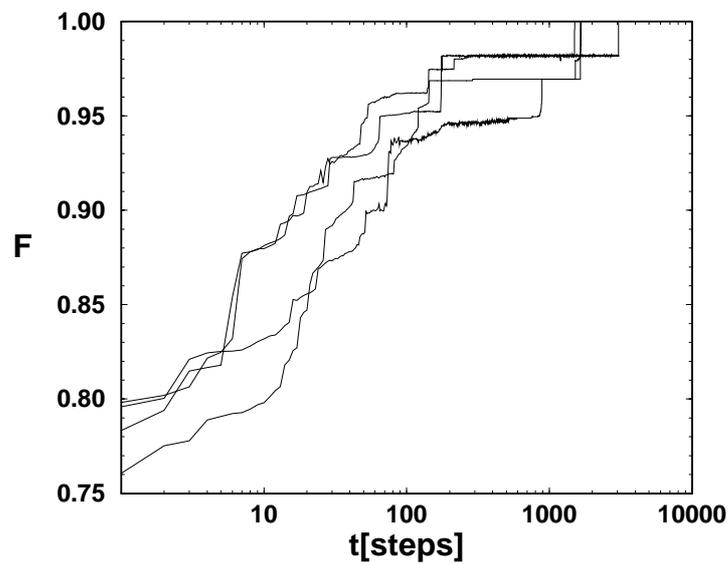


Abbildung 3.14: Das Zeitverhalten der Fitness  $F$  des besten Individuums in einem evolutionären Algorithmus für verschiedene Anfangsbedingungen. Lange Perioden der Stagnation werden von plötzlichen Sprüngen unterbrochen.

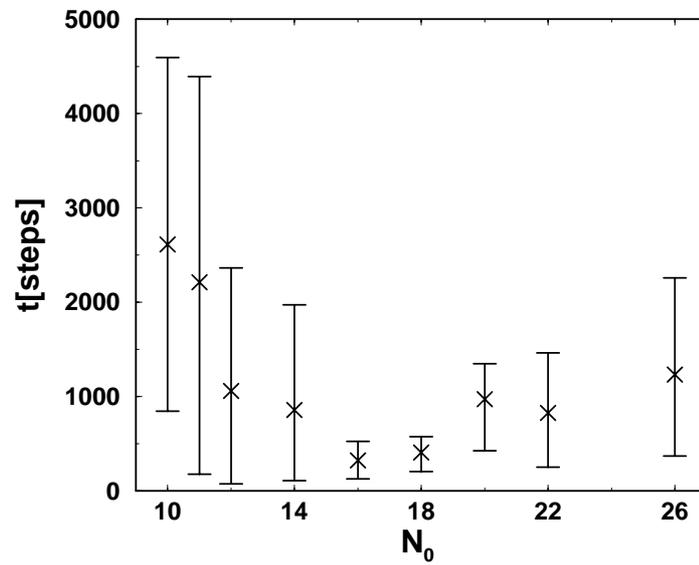


Abbildung 3.15: Die Anzahl der benötigten Optimierungsschritte  $t$  um die optimale Lösung zu finden als Funktion der Zahl der Lampen pro Set  $N_0$  in der Startkonfiguration. Die Rechenzeit, die proportional der Zahl der Iterationen ist, fällt zunächst mit wachsendem  $N_0$  und erreicht ihr Minimum bei  $N_0 = 16$ . Für größere  $N_0$  wächst die Rechenzeit schwach.

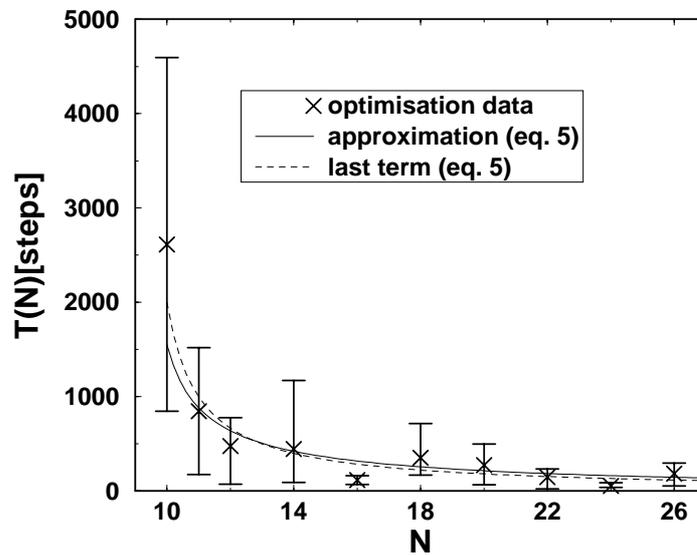


Abbildung 3.16: Die Anzahl der Optimierungsschritte  $T_N$  über der Anzahl der Lampen pro Set  $N$ . Die gemittelten Daten der Simulationen sind mit Kreuzen und Fehlerbalken gezeichnet. Die durchgezogene Linie und die gestrichelte Linie zeigen die Abschätzungen entsprechend Gleichung (3.50) und den letzten Summanden in der Summe (3.50). Die Zeit  $\tau$  diente als Fitparameter.

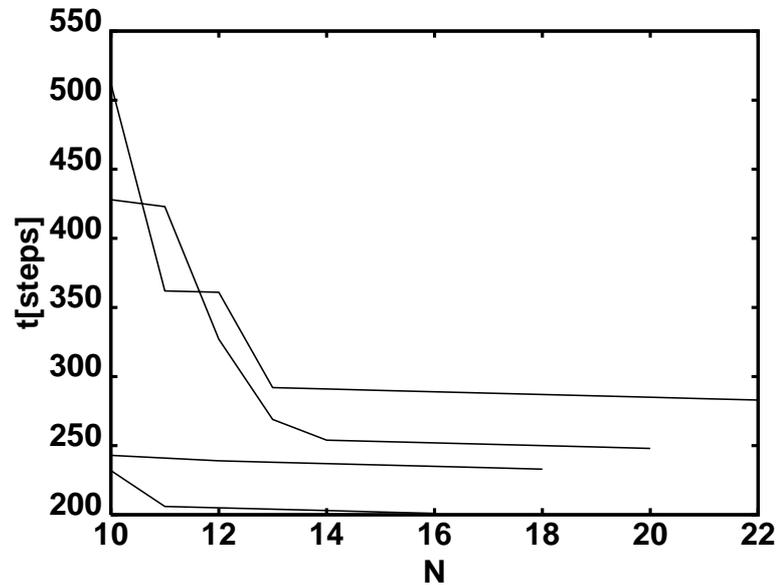


Abbildung 3.17: Einige typische Optimierungsläufe. Nachdem die Lösung für  $N_0 > N_{min}$  Lampen gefunden ist findet der Algorithmus schnell die Lösung für kleinere  $N$ . In Übereinstimmung mit Abbildung 3.15 wird der Algorithmus etwas uneffektiver, wenn man mit großen  $N_0$  startet (Erklärung s. Text).

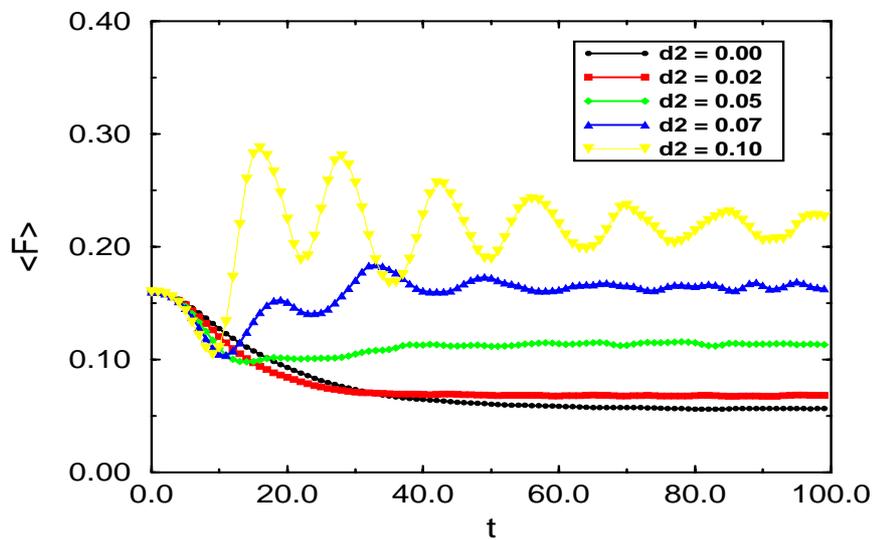


Abbildung 3.18: Zeitliche Entwicklung der mittleren Fitness  $\langle F \rangle$  für verschiedene Werte des Energieübertrags  $d_2$  (Parameter:  $q_0 = 1.0, c = 0.1, \gamma = 0.5, D = 0.025$ )

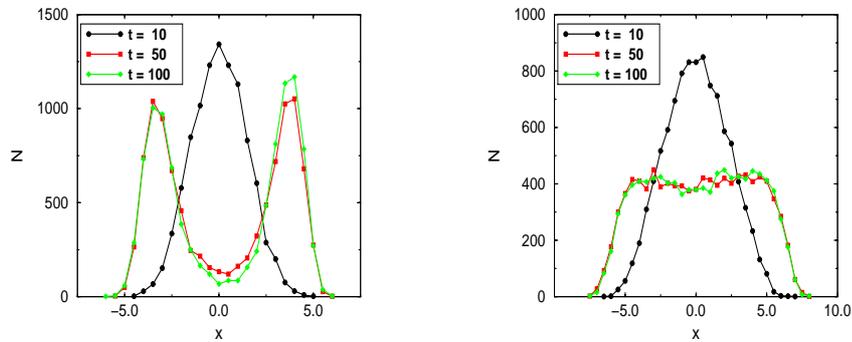


Abbildung 3.19: Verteilung des Ortes  $x$  bei verschiedenen Zeiten  $t$ , Anzahl der Teilchen  $N$ ,  $d_2 = 0.0$  (linkes Bild),  $d_2 = 0.06$  (rechtes Bild)

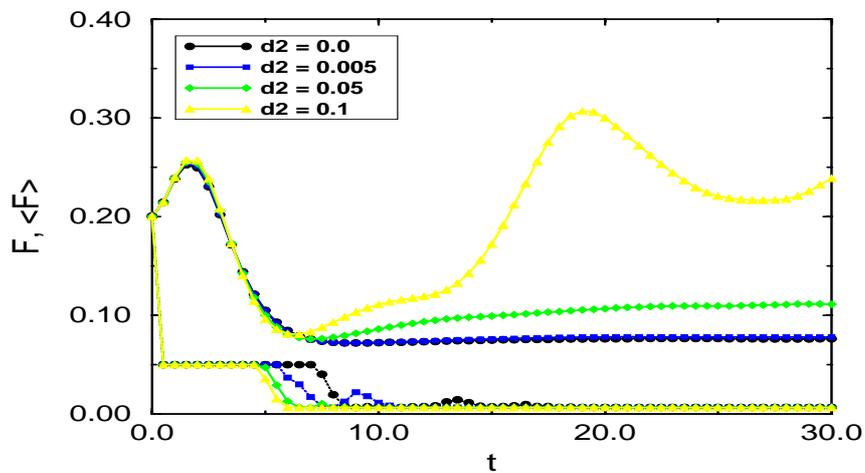


Abbildung 3.20: Zeitliche Entwicklung der kleinsten Fitness  $F$  (gepunktete Linie, untere Linienschar) und der mittleren Fitness  $\langle F \rangle$  (durchgezogene Linie, obere Linienschar) für verschiedene Werte des Energieübertrags  $d_2$  (Parameter:  $q_0 = 1.0, c = 0.8, \gamma = 10.0, D = 10.0$ )



# Kapitel 4

## Strukturelle Optimierung

### 4.1 RNA Sekundärstrukturen

(H. Rosé)

Alle lebenden Organismen besitzen als grundlegende Bausteine ihrer Existenz hochorganisierter Makromoleküle. Diese sind im Stande, in Abhängigkeit ihrer Struktur, die verschiedensten lebenswichtigen Funktionen zu übernehmen. Es war also gerade die Struktur dieser Moleküle, die es im evolutionären Prozeß zu optimieren galt. Welche konkrete räumliche Struktur ein Biomolekül annehmen wird, hängt entscheidend von der Aneinanderreihung seiner elementaren Bausteine ab. Will man den Prozeß ihrer Entstehung verstehen oder gar selbst Biopolymere entwerfen, die bestimmte biologisch aktive bzw. enzymatische Eigenschaften besitzen, hat man die Frage zu beantworten: In welche Struktur wird sich eine gegebene Bausteinsequenz anordnen? Dieses *Faltungsproblem* ist die zentrale Frage der heutigen Biopolymerforschung. Einer der wichtigsten Gegenstände der Untersuchung sind dabei RNA-Sequenzen<sup>1</sup>, denn sie stellen einerseits noch recht einfache Strukturen dar, besitzen andererseits jedoch bereits reichhaltige enzymatische Eigenschaften. Am wichtigsten ist jedoch die Tatsache, daß es möglich ist, sie gezielt, im Hinblick auf ihre Eigenschaften, im Labor herzustellen [Sym92, BJ92].

Jede bestimmte Nukleotidsequenz faltet sich in eine mehr oder weniger stabile räumliche Struktur. Da der Gesamtprozeß der Faltung äußerst kompliziert und nur für recht kurze Sequenzen mit einem vertretbaren Rechenaufwand modellierbar ist, stellte sich die Frage, ob die wichtigsten Eigenschaften von RNA-Strukturen bereits durch ein einfaches Modell beschrie-

---

<sup>1</sup>Ribonukleinsäure-Sequenzen

ben werden können. Dies ist in der Tat möglich und beruht auf dem Erkenntnis, daß die Sekundärstruktur einer RNA-Sequenz in hohem Maße ihre biochemischen Eigenschaften mitbestimmt. Die Sekundärstruktur ist dabei die zweidimensionale Form, die durch die Bindung kompatibler Nukleotide der Sequenz entsteht. Dieser erste Schritt des Faltungsprozesses liefert den Hauptanteil der freien Energie des RNA-Moleküls und bestimmt so seine thermodynamische Stabilität. Der Faltungsprozeß kann damit als Formation von Sekundärstrukturen modelliert werden, die die freie Energie minimieren [FSBB<sup>+</sup>93]. Allerdings ist dabei zu beachten, daß die reale Formation des Biopolymers einen weitaus komplizierten und insbesondere einen dynamischen Prozeß darstellt, der nicht notwendigerweise in einer Struktur minimaler freier Energie resultiert. Es zeigt sich vielmehr, daß es gerade die suboptimalen Sekundärstrukturen sind, die biologische Relevanz besitzen und die Kenntnis ihrer Zustandsdichte wichtig für das Verständnis der Vielfalt und Stabilität der RNA-Moleküle ist [Hig95]. Im Hinblick auf das zu formulierende Optimierungsproblem ist es daher wichtig zu bemerken, daß wir neben der Struktur minimaler freier Energie - also dem Optimum des Problems - vor allem an den Suboptima und der Zustandsdichte interessiert sind. Sie stellen in diesem Falle die eigentlichen Optimierungsziele dar.

Die Zahl möglicher Sekundärstrukturen wächst exponentiell mit der Sequenzlänge  $L$  [FSBB<sup>+</sup>93], wodurch ein komplettes Durchmustern aller Strukturen schon für  $L > 40$  schnell an seine Grenze stößt. Allerdings kann die Zustandsdichte durch eine problemspezifische Methode der Dynamischen Programmierung auch für längere Sequenzen ( $L = 76$ ) bestimmt werden [CHS96, CFRS97]. Diese liefert gleichzeitig jedoch nicht die explizite Gestalt der Sekundärstrukturen. Um das gewünschte Ziel der Bestimmung der suboptimalen Strukturen und der Zustandsdichte zu erreichen, bietet sich die Verwendung evolutionärer Algorithmen gerade zu an. Mit der in Abschnitt 2.3.1 beschriebenen Methode, sind wir in der Lage, sowohl die Suboptima als auch die Zustandsdichte in einem Lauf zu ermitteln. Wir haben dazu kaum mehr zu tun, als das Optimierungsproblem zu formulieren.

Der Suchraum des Faltungsproblems besteht aus allen Sekundärstrukturen, die zur gegebenen Nukleotidsequenz kompatible Bindungspaare aufweisen, d.h., die nur an solchen Positionen der Sequenz Bindungen besitzen, die Watson-Crick-Paare **AU**, **GC** oder das Paar **GU** bilden. Die dabei möglichen Sekundärstrukturen können dabei selbst als Symbolsequenzen aus offenen und geschlossenen Klammern und Punkten dargestellt werden. Die Klammerpaare repräsentieren gebundene und die Punkte ungebundene Nukleotide, also z.B.

GCCCCACGUGGCGGCUA  
 (((((.....))))).

Als Mutationsoperatoren wählen wir die folgenden Operationen:

<b>Binden</b>	..... → .(.....)...
<b>Lösen</b>	.(.....)... → .....
<b>Verschieben</b>	.(.....)... → ..(.....)...
<b>Zuziehen</b>	...(...)... → .(((...)))
<b>Aufziehen</b>	(((...))) → ...(...)...

Die ersten drei Operationen stellen offensichtliche Elementarmutationen dar, während die letzten zwei korrelierte Mehrfachanwendungen der ersten beiden Mutationen sind. Ihre Hinzunahme erwies sich als entscheidend, da gerade aufeinander folgende Paarungen (stacks) hohe negative Energiebeiträge liefern, andererseits bewirken Lücken in dieser Bindungsfolge (inner loops) große Energieerhöhungen. Ein unkorrelierter Aufbau der stacks würde unweigerlich inner loops produzieren, deren Entstehen im Optimierungsprozeß sich als wiederholte Verschlechterung der Fitness darstellen würde. Die Wahrscheinlichkeit des Entstehens längerer stacks wäre damit faktisch Null. Gerade sie stellen aber das typische Strukturelement der gesuchten Optima dar. Die Forderung der Ergodizität des Suchprozesses legt also die Einführung der letzten beiden Mutationsoperatoren nahe.

Die Fitnessfunktion des Minimierungsproblems ist durch die freie Energie der Sekundärstruktur gegeben, die durch eine Unterroutine des Standardsimulationspaketes VIENNA RNA package 1.1 beta für eine gegebene Struktur und Sequenz leicht bestimmt werden kann<sup>2</sup> [HFS<sup>+</sup>94].

Als RNA-Sequenz wurde der erste Teil (5'-cloverleaf,  $L = 100$ ) des Polio Virus Type 1 Mahoney (AC V01148) gewählt, der als Bestandteil der Steuersequenzen des Virus von besonderen biologischen Interesse ist. Die Anwendung der Boltzmannstrategie lieferte sowohl die Sekundärstruktur minimaler freier Energie  $F = -28.09$  kcal/mol, die auch durch den Faltungsalgorithmus des VIENNA RNA package gefunden wird. Darüber hinaus findet die Boltzmannstrategie jedoch auch suboptimale Faltungsstrukturen (s. Abb. 4.1) und die Zustandsdichte (Abb. 4.2) durch die Anwendung der Methode aus Abschnitt 2.3.1.

Wie erwartet besitzt auch die Zustandsdichte des Polio Virus im positiven Energiebereich einen gaußschen Verlauf, da die Sekundärstrukturen hoher positiver Energie sich aus vielen nahezu unabhängigen Subelementen

---

<sup>2</sup>Dabei wird die gesamte Sekundärstruktur in Substrukturen zerlegt, deren Energiebeiträge durch Labormessungen bestimmt werden konnten. Diese bilden in der Summe die freie Energie der Sekundärstruktur.

zusammen setzen. Dies stimmt mit den Ergebnissen überein, wie sie für Hefe in [CFRS97] gefunden wurden.

## 4.2 Das LABS-Problem II

(D. Beule, B. Militzer, M. Zamparelli)

Die Suche nach binären Sequenzen mit kleinen Autokorrelationsfunktionen (LABS) stellt ein mit steigender Länge der Sequenzen extrem schnell schwieriger werdendes Problem dar. Solche Sequenzen werden z. B. in der Signalverarbeitung benötigt [Ber87]. Man betrachtet binäre Sequenzen  $(s_1, \dots, s_L)$  der Länge  $L$ , ( $s_i = \pm 1$ ) und ihre Energie

$$E = \sum_{k=1}^{L-1} R_k^2 \quad \text{mit} \quad R_k = \sum_{i=1}^{L-k} s_i s_{i+k} \quad (4.1)$$

aus der sich der *Merit-Faktor* ableiten läßt

$$F = \frac{L^2}{2E}, \quad (4.2)$$

der für unsere Zwecke die *Fitness* einer Sequenz darstellt, die zu maximieren ist. Die Zahl der möglichen Sequenzen beträgt  $2^L$ , so daß eine vollständige Suche für  $L > 40$  unpraktikabel ist.

LABS entspricht außerdem der Suche nach dem Grundzustand in einem Ising-Spinsystem mit langreichweitiger 4-Spin-Wechselwirkung [BM94] und ist ein klassisches frustriertes Optimierungsproblem, welches von verschiedenen Autoren untersucht wurde [Ber87, Dit96, BM94, Rud95].

Zum ersten Mal wurden für dieses Problem  $(\mu, \lambda)$  Strategien, zusammen mit einem Mutationsoperator, der sich von den normalen bisher angewandte Einzel-flip wesentlich unterscheidet, angewandt. Dieser Mutationsoperator nutzt das vorhandene Wissen durch ein Vorauswahlprozeß in dem mehrere mögliche Konfigurationen betrachtet werden, aus. Das erfolgt durch die Auswahl von  $r$  verschiedenen  $R_k$ 's mit überdurchschnittlich hohen Werten und durch das Flippen von  $b$  Bits bis alle  $r$  Korrelationen kleiner geworden sind. Damit wird ein zielgerichteter und nicht mehr rein zufällig Mutations-schritt gewonnen. Auf diese Weise wurden für alle in der Literatur bisher behandelten Werte von  $L$  die Rekorde erreicht oder übertroffen.

Insbesondere für  $L = 100$  wo keine die Dimension halbierende Symmetrie (Skew-symmetry) ausgenutzt werden kann, wurden Sequenzen gefunden, die großordnung 10 mal weniger häufig sind, als die mit normalen 1-Bit Flips und Boltzmann Strategien gefundenen.

## 4.3 Optimierung von Straßennetzen

(H. Rosé)

Netzwerkoptimierung gehört zu der Gruppe der Probleme, die derzeit immer mehr ins Zentrum der Beachtung rücken, denken wir an Anwendungen wie Daten- und Kommunikationsnetze oder Stadtplanung. Letzteres war Gegenstand der Untersuchung des SFB230 zu dem im Projekt Kooperationsverbindungen bestanden. Das von Projektpartner SIEMENS München ebenfalls geäußerte Interesse an dieser Problemstellung machte die Netzwerkoptimierung zu einer relevanten Anwendungsaufgabe, für die im Projekt entwickelten gemischten Strategien (s. 3.3).

Die einfachste Problemstellung der Netzwerkoptimierung besteht in der Minimierung der Netzwerklänge - oder damit verbunden der Netzwerkaufbaukosten. Diese ist jedoch sowohl für praktische Belange als auch für das theoretische Interesse von nur geringen Gehalt. Nimmt man jedoch die Forderung der gleichzeitigen Minimierung des mittleren Umwegs (Abweichung der Länge des Weges zwischen zwei Punkten im Netz zur direkten Distanz zwischen beiden gemittelt über alle Punktpaare) hinzu, so erhält man ein bis dahin nichtuntersuchtes und praktisch relevantes Optimierungsproblem.

Es gehört zur Klasse der frustrierten Probleme, da die Netzlänge bei kleiner (größer) werdenen Umweg wächst (schrumpft), beide Ziele also nie gleichzeitig minimal werden können. Die optimale Lösung ist ein Kompromiß zwischen Länge und Umweg, zwischen Kosten und Komfort oder zwischen Bau- und Fahrtzeit. Um dennoch eine Qualität eindeutig definieren zu können muß eine Zielsetzung in Form einer Gewichtung der widersprechenden Forderungen erfolgen. Im einfachsten Falle kann dies durch einen Preisparameter  $\lambda$  geschehen, der die Kosten pro Netzwerklänge festlegt. Die Fitneß eines Netzes  $g$  wird dann durch

$$F(g) = \lambda c(g) + (1 - \lambda) d(g), \quad 0 \leq \lambda \leq 1 \quad (4.3)$$

beschrieben, wobei  $c(g)$  die Länge des Netzes und  $d(g)$  den mittleren Umweg, darstellen [SERW96].

Für maximalen Preis  $\lambda = 1$  erhalten wir das einfache Längenminimierungsproblem, dessen optimale Lösung der minimal vernetzte Graph ist und der durch bekannte mathematische Algorithmen bestimmt werden kann. Für den kostenlosen Bau eines Netzes  $\lambda = 0$  ist der vollvernetzte Graph mit  $F(g_v) = d(g_v) = 0$  die optimale Lösung. Für alle realen Situationen  $0 < \lambda < 1$  stellt die Lösung einen Kompromiß zwischen diesen beiden Grenzfällen dar.

Der Suchraum des Problem ist die Menge aller zusammenhängender Graphen mit  $N$  Knotenpunkten. Die Zahl der möglichen Graphen ist unvorstell-

bar groß<sup>3</sup>. Als Punktconfiguration verwenden wir die 39 wichtigsten Knotenpunkte des Straßennetzes der italienischen Kleinstadt Martina Franca (s. Abb. 4.4).

Zur Optimierung wenden wir die im Projekt entwickelte gemischte Strategie (s. 3.3) und zum Vergleich die Boltzmann Strategie (s. 3.1) an. Abbildung 4.5 zeigt den Verlauf der Boltzmann Optimierung für  $\lambda = 0.975$ . Ausgehend vom vollvernetzten Graphen beginnt der Algorithmus zunächst das Netzwerk auszudünnen. Dies ist gut in Abbildung 4.6 zu erkennen, die die Entwicklung der Netzwerklänge zum Umweg wiedergibt. Mit schrumpfender Länge vergrößert sich zunächst der Umweg. Nachdem der Großteil überflüssiger Verbindungen entfernt ist, beginnt der Algorithmus von sich aus den Graphen zu balancieren (Abb. 4.5, Bild 3,4). Dies führt nun auch zur Minimierung des Umwegs (Abb. 4.6, linke Hälfte). Nach  $6 \cdot 10^5$  Schritten stagniert die Boltzmann Strategie schließlich in einem lokales Optimum der Fitneß.

Die Anwendung der gemischten Strategie erzielt im Vergleich zur Boltzmann Strategie (jeweils 16 Sucher) bereits nach einem Sechstel der Zeit ein besseres Resultat. Abbildung 4.7 zeigt die Zeitentwicklung der Fitneß und die optimierten Netzwerke für beide Strategien. Die gemischte Strategie minimiert deutlich schneller die Fitneß und findet ein sichtbar besseres Netzwerk (Abb. 4.7, rechts).

Im Resultat können wir feststellen, daß die gemischte Strategie im Falle der Netzwerkoptimierung eindeutig zu bevorzugen ist. Nicht nur ihre schnellere Konvergenz, sondern vor allem ihre Fähigkeit lokale Optima zu verlassen und damit den Weg hin zum globalen Optimum weiter zu verfolgen, prädestiniert sie für die praktische Anwendung.

## 4.4 Adaptive Netzwerkoptimierung

(M. Zamparelli)

Eines der schwierigsten Probleme im Bereich der Wegenetzoptimierung ist das sogenannte Steiner Tree Problem [HRW92]. Die Aufgabe besteht darin, vorgegebene Punkten (Städte), unter Einführung zusätzlicher Knoten so zu verbinden daß die gesamte Länge des Netzwerkes minimal wird. Genauer gesagt die Anzahl, Koordinaten und Verbindungsmatrizen der zusätzlicher Knoten mußten festgestellt werden.

---

<sup>3</sup>Die Anzahl aller Graphen über  $N$  Punkten beträgt  $2^{N(N-1)/2}$ , z.B. für  $N = 39$   $10^{223}$ .

Von praktischem Interesse abgesehen (Telefonnetze oder Bahn) ist dieses Probleme auch für Theoretiker von Bedeutung, da es zu der Klasse der außerordentlich schwierigen NP-complete Probleme [GGJ79] zählt, für die neue Heuristiken immer sehr erwünscht sind. Mit Hilfe einer hierarchischen Aufbau-prozedur durch ein Abbildung zwischen der Menge der Listen (siehe Fig. 4.4) mit  $N$  Blättern und der Menge aller zusammenhängende nicht azyklische Graphen (Bäume) der Suchraum wesentlich eingeschränkt wird.

Diese Abbildung erlaubt ohne Näherungsverfahren einen kontinuierlichen Suchraum durch einem diskreten zu ersetzen. Die Aufgabe ist daher neu formuliert indem man aus dem Raum aller möglichen Listen, die mit dem entsprechenden minimalen Graph sucht.

Die Leistung dieses Algorithmus für bis  $N = 30$  ist vergleichbar mit anderen Heuristik-basierten Methoden, allerdings ist abzusehen, daß für größere Werte von  $N$  die Kausalität gebrochen wird, und kleine Änderungen in der Liste zu große Änderungen in der Fitness führen.

## 4.5 Erstellte Software

(H. Rosé)

### 4.5.1 Wea

**NAME:** wea

**BESCHREIBUNG:**

Das Programm implementiert die gemischte und die Self Annealing Strategie zur Netzwerkoptimierung (s. 4.3) unter Verwendung der Bibliothek *ArtEvo* (s. 3.10.1). Über ein Koordinatenfile können beliebige Punkt-konfigurationen (z.B. martina.crd) eingelesen werden. Folgende Betriebsarten wurden implementiert:

- *density*: Bestimmung der Zustandsdichte mittels Boltzmannstrategie.
- *mixed*: Optimierung durch die gemischte Strategie.
- *adaptive*: Optimierung durch die Self Annealing Strategie.

**PARAMETER:** Parameterfile: wea.para

Populationsize Populationsgröße  
Nodenummer Punktzahl des Graphen  
Generations Generationen  
Lambda Preisfaktor der Fitneßfunktion  
Fillfactor Startvernetzungsgrad  
Temperature Starttemperatur  
Annealing Abkühlungsfaktor (falls kein Self Annealing)  
Tvariance Starttemperaturvarianz  
Mutation Mutationswahrscheinlichkeit  
Selection Selektionswahrscheinlichkeit  
Coordfile Punktkoordinatenfile  
Writestep Speicherperiode  
Mode Programmmode  
Addnoise (0 = multiplikatives Rauschen)

**AUTOR:** Helge Rosé (rose@summa.physik.hu-berlin.de)



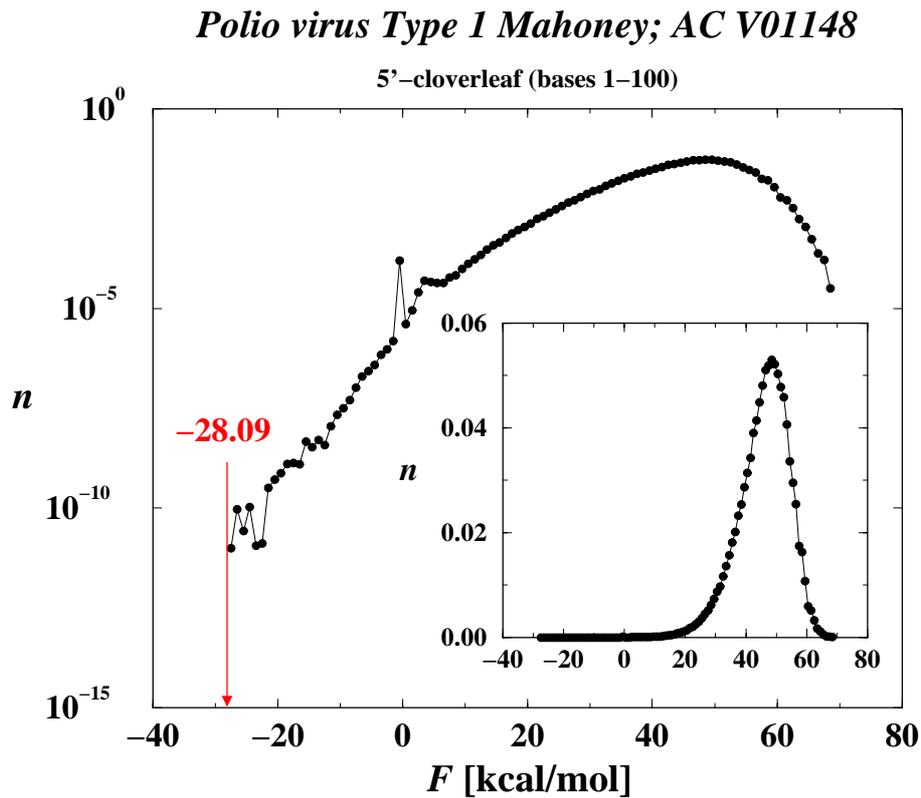
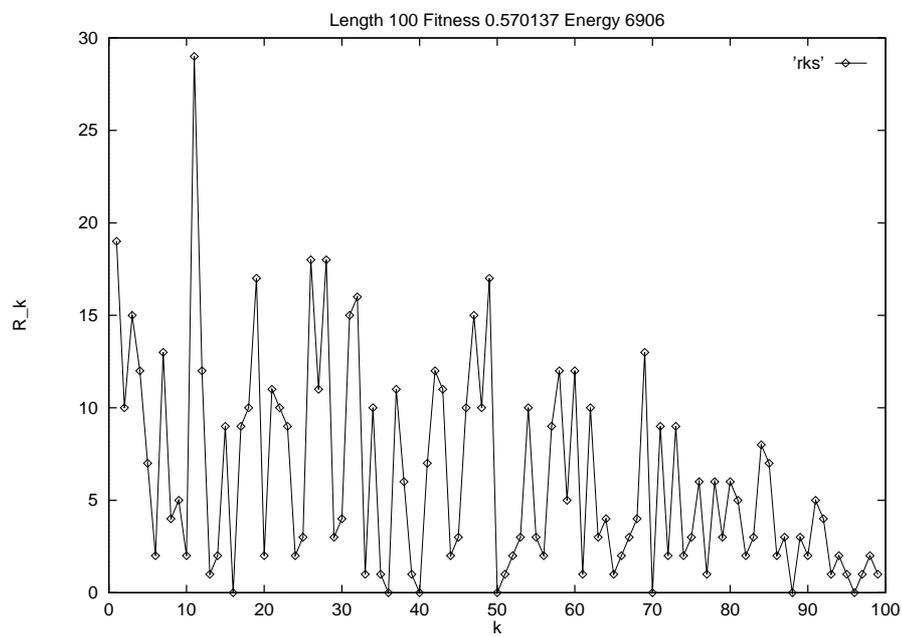


Abbildung 4.2: Zustandsdichte des ersten Teils ( $L = 100$ ) des Polio Virus Typ 1 Mahoney (AC V01148), die durch eine Boltzmannstrategie bestimmt wurde. Zu Verdeutlichung des nahezu gaußschen Verlaufs im positiven Energieintervall, ist die Dichte neben der semilogarithmischen auch in linearer Darstellung wiedergegeben.

Abbildung 4.3: Beträge der  $R_k$  für eine Sequenz mit niedrigem Merit-Faktor

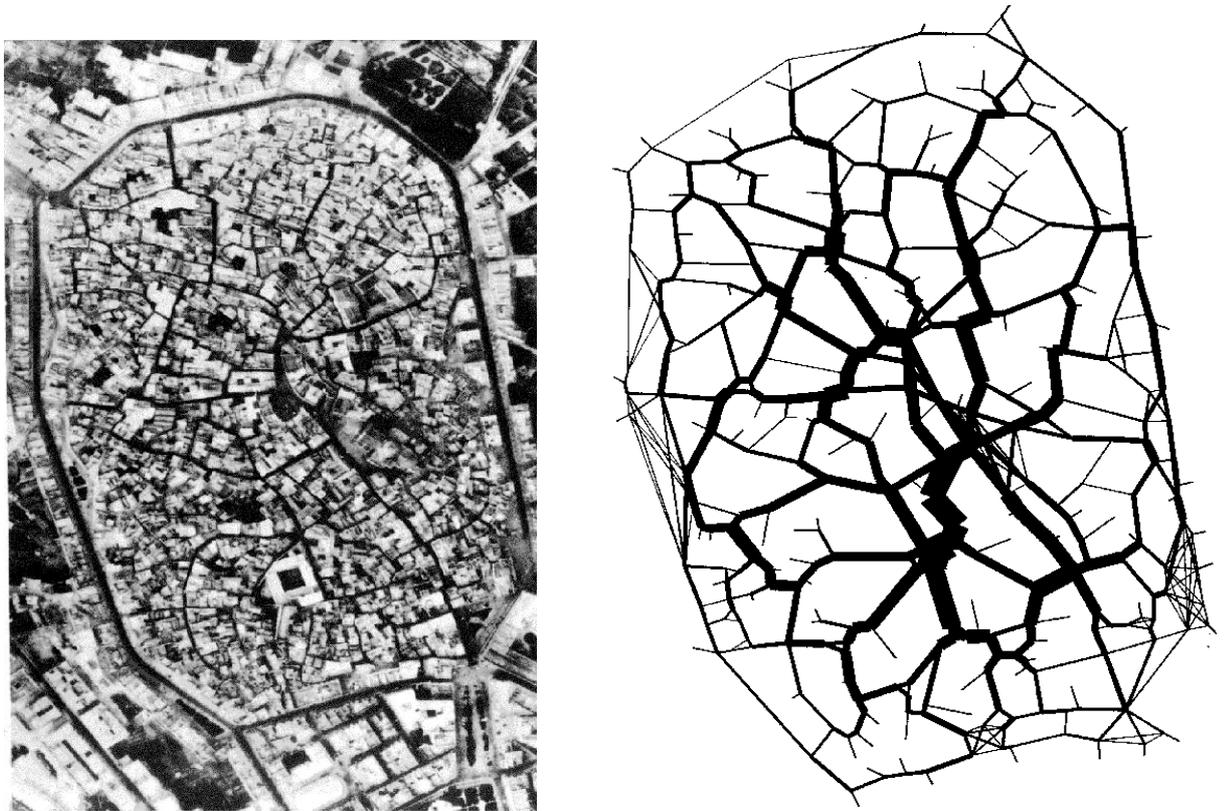


Abbildung 4.4: Straßennetz der italienischen Kleinstadt Martina Franca.

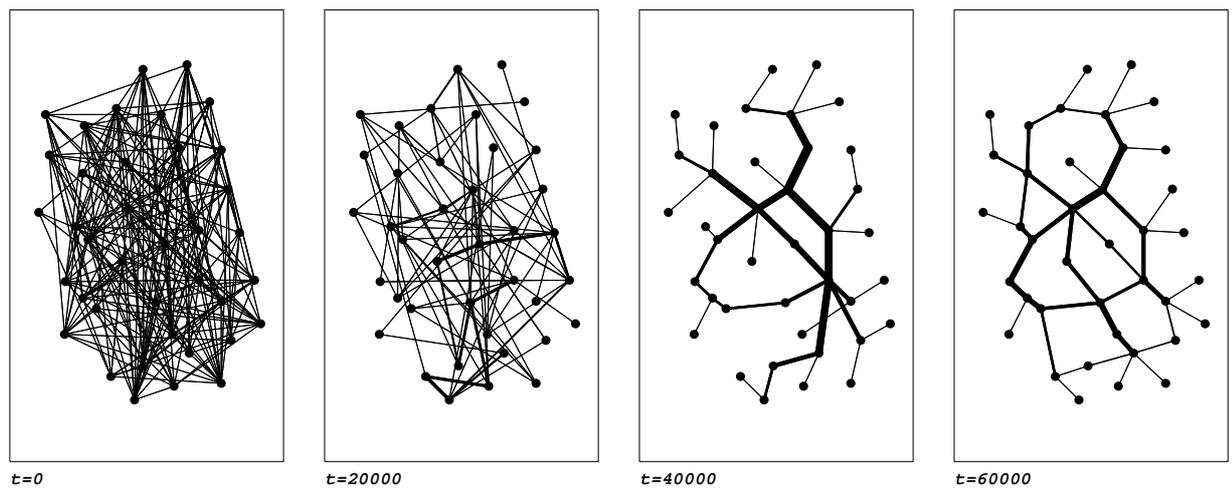


Abbildung 4.5: Verlauf der Optimierung mittels Boltzmann Strategie.

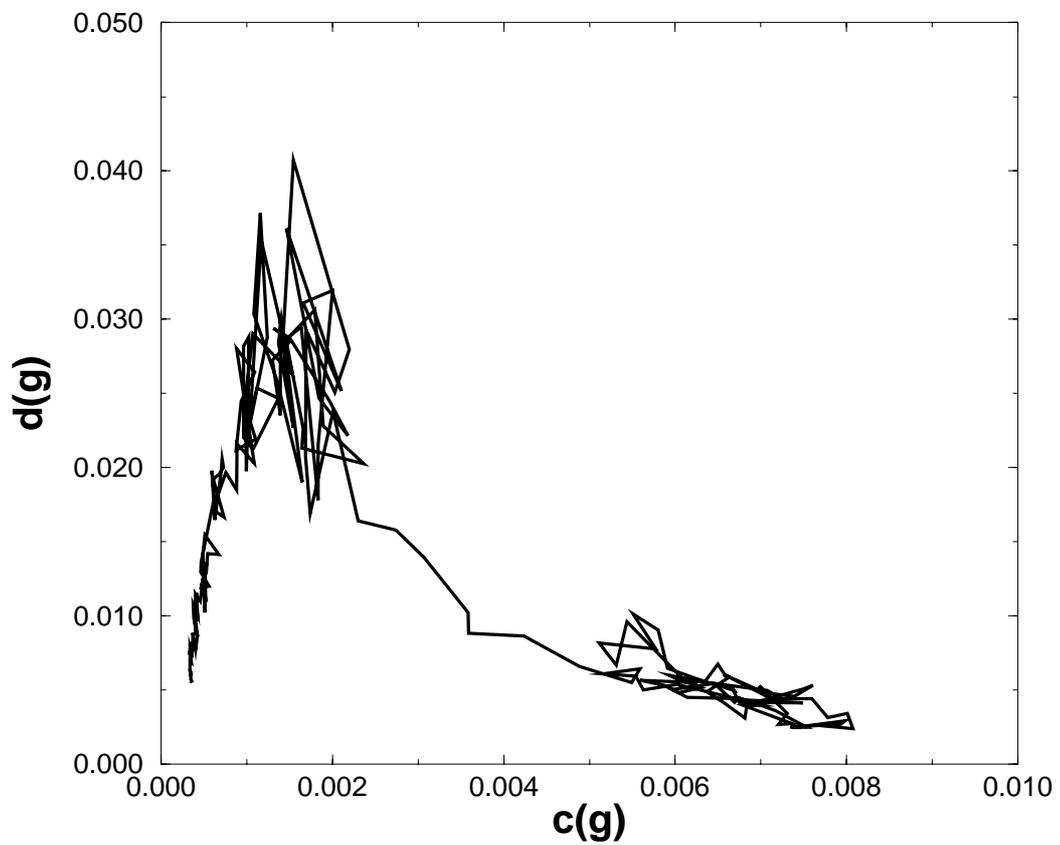


Abbildung 4.6: Entwicklung der Netzwerklänge  $c$  zum Umweg  $d$ . Der Optimierungsprozeß startet in der rechten unteren Ecke.

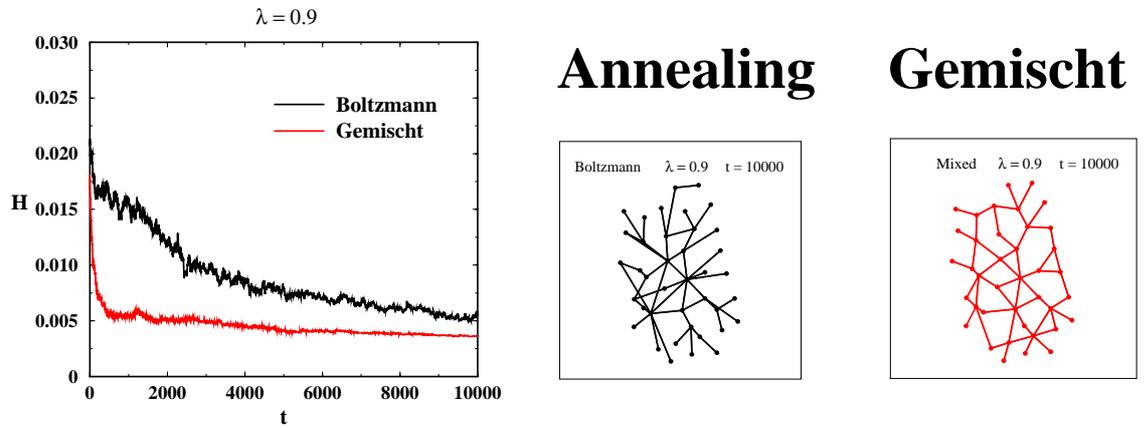


Abbildung 4.7: Vergleich der Boltzmann und der gemischten Strategie,

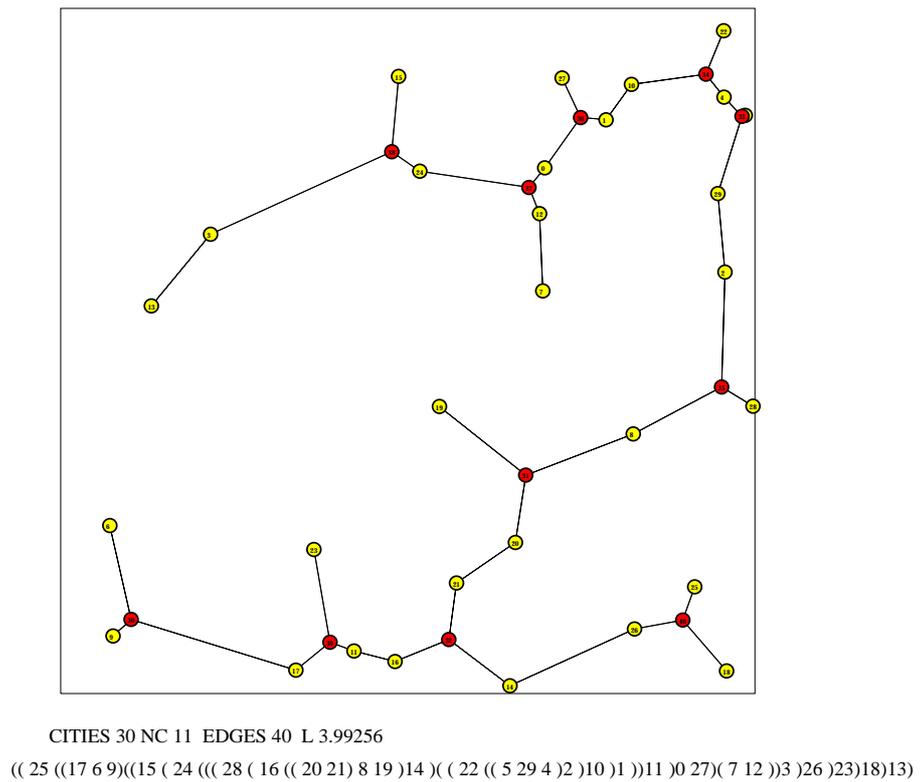


Abbildung 4.8: Die Ergebnisse des Algorithmus für  $N = 30$

# Kapitel 5

## Anwendungen auf Probleme der Physik

### 5.1 Das Beladeproblem von Druckwasserreaktoren

(C. Kappler, M. Zamparelli)

Wegen des vorzeitigen Ausscheidens der beiden Bearbeiter ohne ordnungsgemäße Berichterstattung muss in Bezug auf diesen Teil auf die vorliegenden Publikationen Kappler/Zamparelli et al. verwiesen werden.

### 5.2 Molekulardynamik mit Cauchyverteilungen

(F. Schautz)

Es wurden mit einer neu entwickelten Methode Molekulardynamik - Simulationen des Elektronengases durchgeführt. Sie besteht darin, daß in einer ersten Phase eine Optimierungsmethode (Simulated Annealing) eingesetzt wurde, um die relevanten Anfangsbedingungen für die Simulationen zu finden, D. h. solche Anfangsbedingungen, die im Gleichgewicht hohe Phasendichte haben. Damit wird in der zweiten Phase die eigentliche Simulation gestartet. Insbesondere Beispiel wurde ein Elektronengas simuliert, in dem quantenmechanische Effekte (Pauliprinzip) durch zusätzliche *impulsabhängige* Potentiale zwischen den Elektronen berücksichtigt sind. Eine Folge der impulsabhängigen Wechselwirkungen ist, daß herkömmliche (d.h. für Systeme

mit ausschließlich ortsabhängigen Wechselwirkungen entwickelte) Methoden das thermodynamische Gleichgewicht zu erreichen nicht angewendet werden konnten. Stattdessen wurde die erste Phase durch Simulated Annealing ersetzt, was das Einstellen des Gleichgewichtes unabhängig von der Gestalt der Wechselwirkungen zwischen den Teilchen ermöglicht. Dabei wird ein Random Walk durch den Phasenraum konstruiert, bei dem einzelne Phasenraumpunkte entsprechend ihrer Wahrscheinlichkeit im thermodynamischen Gleichgewicht angelaufen werden. Damit wird erreicht, daß (im Limes unendlich langer Simulationszeit) entlang dieses Random Walk berechnete Mittelwerte (z.B. der Energie) mit denen im thermodynamischen Gleichgewicht übereinstimmen. Da die Simulationen in der Praxis in endlicher Zeit durchgeführt werden müssen, ist es wichtig, den Phasenraum effektiv abzusuchen. D. h. es tritt dasselbe Problem auf wie in der hochdimensionalen Parametereoptimierung durch evolutionäre Algorithmen (s. o.). In diesem Fall wurde der Phasenraum durch koordinatenweises Ziehen nach einer Cauchyverteilung abgesucht. Durch dieses Verfahren wurde das thermische Gleichgewicht deutlich schneller erreicht als durch Ziehen der Variablen nach einer Gaußverteilung.

### 5.3 Die Anwendung evolutionärer Algorithmen auf quantenmechanische Probleme

#### (D. Rosenkranz)

Viele quantenmechanische Aufgaben lassen sich analytisch nur näherungsweise oder gar nicht lösen. Es existieren allerdings verschiedenste numerische Methoden, die für die unterschiedlichen Problemgebiete mehr oder minder geeignet sind. Ziel der vorliegenden Arbeit war es nun, ausgehend von einem speziellen Evolutionsmodell, dem Fisher–Eigen–Modell, eine numerische Lösungsmethode für verschiedene quantenmechanische Problemstellungen zu entwickeln und zu testen.

Die Evolution einer Population der Dichte  $\rho(\vec{x}, t)$  in einem  $d$ -dimensionalen Phasenraum mit einer Fitneßlandschaft, die durch die skalare Funktion  $E(\vec{x})$  gegeben ist, wird durch die Fisher–Eigen–Gleichung beschrieben [ESG88, EEF90b]:

$$\frac{\partial}{\partial t} \rho(\vec{x}, t) = D \frac{\partial^2}{\partial \vec{x}^2} \rho(\vec{x}, t) + (E(\vec{x}) - \langle E \rangle) \rho(\vec{x}, t), \quad (5.1)$$

### 5.3. DIE ANWENDUNG EVOLUTIONÄRER ALGORITHMEN AUF QUANTENMECHANISCHE

wobei der Mittelwert der Fitneß  $\langle E \rangle$  als normaler Ensemblemittelwert definiert ist.

Die numerische Repräsentation der Dichte  $\rho(\vec{x}, t)$  durch ein hinreichend großes Ensemble Brownscher Teilchen führt auf eine einfache Methode, Gl.(5.1) auf dem Computer orts- und zeitabhängig zu lösen (für eine ausführliche Darstellung siehe [Ros96b] und [Fri94]).

#### Quantenmechanischer Grundzustand

Die Schrödingergleichung

$$-i\hbar \frac{\partial}{\partial t} \phi(\vec{x}, t) = \frac{\hbar^2}{2m} \frac{\partial^2}{\partial \vec{x}^2} \phi(\vec{x}, t) - V(\vec{x}) \phi(\vec{x}, t) \quad (5.2)$$

kann durch Einführen einer imaginären Zeit ( $t \rightarrow -it$ ) und mit Hilfe des Ansatzes

$$\phi(\vec{x}, t) = \exp \left( - \int_0^{t/\hbar} \langle V \rangle dt' \right) \rho(\vec{x}, t) \quad (5.3)$$

in die Fisher–Eigen–Gleichung (5.1) transformiert werden, wobei gilt:  $E(\vec{x}) = -V(\vec{x})$  und  $D = \hbar^2/2m$ . Die Dichte  $\rho(\vec{x}, t)$  konvergiert für lange Zeiten gegen die Grundzustandswellenfunktion des quantenmechanischen Systems, so daß auch der Mittelwert  $\langle V \rangle$  gegen die Grundzustandsenergie konvergiert [EEF90b, Fri94, Ros96b]. Die Konvergenzgeschwindigkeit hängt im wesentlichen von der Energiedifferenz des Grund- vom ersten angeregten Zustand ab.

Als Test haben wir einfache quantenmechanische Systeme, wie den harmonischen Oszillator und das Wasserstoffatom, aber auch kompliziertere 2–Teilchensysteme, wie das Heliumatom oder das negativ geladenen Wasserstoffion untersucht. Es zeigte sich eine gute Übereinstimmung mit vorhandenen analytischen Resultaten (für eine ausführliche Darstellung siehe [Ros96b]).

Die Genauigkeit der numerischen Ergebnisse und besonders der zeitliche und rechentechnische Aufwand zu ihrem Erhalt hängen natürlich stark von der Komplexität des zu untersuchenden Systems und der für jedes Problem spezifischen Wahl der Simulationsparameter ab. Prinzipiell läßt sich diese Methode aber zur Ermittlung des bosonischen Grundzustandes beliebiger quantenmechanischer Systeme einsetzen.

Ein ungelöstes Problem stellt noch die Einbeziehung fermionischer Freiheitsgrade dar; so müßte man die globale Koppelung zwischen Teilchen, die durch eine explizite Symmetrisierung eingeführt wird, in den Hamiltonian

und damit in die Simulationsvorschrift einfügen, was allerdings kein triviales Unterfangen ist.

Eine zur vorgestellten ähnliche Methode benutzen Ceperley et al. [CA80, CA87] unter dem Namen Diffusion–Monte–Carlo, wobei diese Methode allerdings mit zusätzlichen Testwellenfunktionen arbeitet, die für jedes Problem geeignet konstruiert und angepaßt werden müssen. Durch geeignete Konstruktion dieser Testwellenfunktionen lassen sich damit aber auch fermionische Grundzustandsenergien abschätzen.

### Quantenmechanische Dichtematrix

Die Dichtematrix bzw. der Dichteoperator  $\hat{\rho}(\beta)$  (mit  $\beta = (k_B T)^{-1}$ ) ist eine wichtige Größe in der Quantenstatistik, da, wenn sie bekannt ist, ein quantenstatistisches Gleichgewichtssystem vollständig bestimmt ist. Beliebige quantenmechanische Erwartungswerte von Operatoren lassen sich dann berechnen nach:

$$\langle \hat{F} \rangle = \frac{\text{Sp}(\hat{F}\hat{\rho})}{\text{Sp}(\hat{\rho})}. \quad (5.4)$$

Für realistische Quantensysteme ist dieser Operator analytisch jedoch schwer zu fassen.

In der Ortsdarstellung gilt für die Dichtematrix  $\rho(\vec{x}, \vec{x}', t)$  folgende Evolutionsgleichung, die Blochgleichung [Fey72]:

$$\frac{\partial}{\partial \beta} \rho(\vec{x}, \vec{x}', \beta) = -H_{\vec{x}} \rho(\vec{x}, \vec{x}', \beta) = \left( \frac{\hbar^2}{2m} \frac{\partial^2}{\partial \vec{x}^2} - V(\vec{x}) \right) \rho(\vec{x}, \vec{x}', \beta) \quad (5.5)$$

mit der Anfangsbedingung  $\rho(\vec{x}, \vec{x}', 0) = \delta(|\vec{x} - \vec{x}'|)$ . Die Variable  $\vec{x}'$  hat in dieser Gleichung nur die Funktion eines Parameters, der die Lokalisierung der Anfangsbedingung bestimmt. Mit Hilfe des Ansatzes (5.3) kann diese Evolutionsgleichung wiederum in die Fisher-Eigen-Gleichung transformiert werden; mit  $\beta$ , der inversen Temperatur, als Systemzeit (genauer:  $t = \hbar\beta$ ) und der Anfangsbedingung  $\rho_{FE}(\vec{x}, \vec{x}', 0) = \delta(|\vec{x} - \vec{x}'|)$ . Um die Dichtematrix zu bestimmen, muß also der Phasenraum bzw. der interessierende Teil des phasenraumes mit den Anfangsbedingungen abgetastet werden und für jede Anfangsbedingung die Zeitentwicklung der Fisher-Eigen-Dichte aufgezeichnet werden. Aus dieser Dichte kann dann entsprechend Gl.(5.3) die unnormierte und aus ihr die normierte Dichtematrix bestimmt werden. Der numerische Aufwand zur Bestimmung der Dichtematrix ist gerade bei hochdimensionalen Phasenräumen immens hoch, prinzipiell sollte aber für bosonische Systeme, bzw. Systeme ohne spezielle Symmetrie diese Methode anwendbar sein.

### 5.3. DIE ANWENDUNG EVOLUTIONÄRER ALGORITHMEN AUF QUANTENMECHANISCHE

Als erste Modellsysteme wurden der harmonische Oszillator, der kubisch gestörte Oszillator, verschiedene Doppelpotentiale und das Morsepotential [LL87] untersucht. Für diese Systeme dauerte die Berechnung der relevanten Diagonalelemente der Dichtematrix auf einer DEC-Alpha Workstation jeweils ca. 3 Stunden. Da viele Operatoren nur von einer Koordinate abhängen, ist für viele Zwecke die Kenntnis der Diagonalelemente bereits ausreichend, so daß sich also der rechentechnische Aufwand für einfache Systeme durchaus in Grenzen hält. Mit Hilfe der so ermittelten Dichtematrix können dann beliebige Erwartungswerte berechnet werden (siehe [Ros96b]).

Neben den eindimensionalen Problemen kann man natürlich auch Probleme mit höherdimensionalen Phasenräumen behandeln. Man bekommt dann aber schnell Probleme mit der Rechenzeit, da das abzutastende Phasenraumvolumen mit wachsender Dimension sehr schnell größer wird; desweiteren potenziert sich natürlich auch der erforderliche Speicherplatz, denn die volle Dichtematrix hat die Dimension  $d_M = 2 \times d_x + 1$ , so daß man bei  $N_x$  räumlichen und  $N_t$  zeitlichen Stützstellen  $N_x^{2d_x} * N_t$  Werte speichern muß. Selbst wenn man sich nur für die Diagonalelemente interessiert, sind es noch  $N_x^{d_x} * N_t$  Werte. Um den Simulationsphasenraum zu reduzieren, muß man sich dann im System vorhandene Symmetrien zu nutze machen.

#### **Zusammenfassung**

Wir haben eine Methode entwickelt und untersucht, die es ermöglicht, verschiedenste Probleme der Quantenmechanik und Quantenstatistik numerisch zu lösen. Hierbei wird ausgenutzt, daß sich quantenmechanische Bewegungsgleichungen (Schrödingergleichung, Blochgleichung) durch geeignete Transformationen in Evolutionsgleichungen vom Fisher-Eigen-Typ überführen lassen. Diese Evolutionsgleichungen lassen sich numerisch relativ einfach behandeln, so daß man eine numerische Lösungsmethode für quantenmechanische Probleme in die Hand bekommt.

Es konnte für verschiedenste Probleme gezeigt werden, daß diese numerische Methode funktioniert und Ergebnisse guter Genauigkeit mit vertretbarem Aufwand liefert.

Die Limitierung der vorgestellten Simulationsmethode liegt auf prinzipieller Seite in der Beschränkung auf bosonische Systeme und auf technischer Seite im hohen Zeitaufwand und Speicherbedarf bei der Simulation von Mehrteilchensystemen bzw. Systemen mit höherdimensionalen Phasenräumen. Die Überwindung dieser Grenzen erscheint möglich, erfordert aber weitere Untersuchungen.



# Literaturverzeichnis

- [AE97] T. Asselmeyer and W. Ebeling. Unified description of evolutionary strategies over continuous parameter spaces. *BioSystems*, 41:167–178, 1997.
- [AEN94] V. S. Anishchenko, W. Ebeling, and A. B. Neiman. *Chaos, Solitons & Fractals*, 4:69, 1994.
- [AER96] T. Asselmeyer, W. Ebeling, and H. Rosé. Smoothing representation of fitness landscapes - the genotype-phenotype map of evolution. *BioSystems*, 39:63–76, 1996.
- [AER97] T. Asselmeyer, W. Ebeling, and H. Rosé. Evolutionary strategies of optimization. *Phys. Rev. E*, 56:1171–1180, 1997.
- [AG93] B. Andresen and J.M. Gordon. Analytic constant thermodynamic speed-cooling strategies in simulated annealing. *Open Sys. and Inf. Dyn.*, 2:1–12, 1993.
- [AG94] B. Andresen and J.M. Gordon. Constant thermodynamic speed for minimizing entropy production in thermodynamic processes and simulated annealing. *Phys. Rev. E*, 50:4346–4351, 1994.
- [Ass96] T. Asselmeyer. Generalized Heat equation and Topology. *eingereicht bei J. Math. Phys.*, 1996.
- [Ati67] M. Atiyah. *K-Theory*. W.A. Benjamin, New York–Amsterdam, 1967.
- [BE88] T. Boseniuk and W. Ebeling. Evolution strategies in complex optimization: The travelling salesman problem. *Syst. Anal. Model. Simul.*, 5:413–422, 1988.
- [BE90] T. Boseniuk and W. Ebeling. Boltzmann-, Darwin- and Heackelstrategies in optimization problems. In *PPSN Dortmund*, 1990.

- [BEE87a] T. Boseniuk, W. Ebeling, and A. Engel. Boltzmann and Darwin strategies in complex optimization. *Phys. Lett.*, **A 125**:307, 1987.
- [BEE87b] T. Boseniuk, W. Ebeling, and A. Engel. Boltzmann and darwin strategies in complex optimization. *Phys. Lett.*, **A 125**:307–310, 1987.
- [Ber87] J. Bernasconi. Low autocorrelation binary sequences: statistical mechanics and configuration space analysis. *J. Physique*, **48**:559, 1987.
- [BJ92] A. Beaudry and G. Joyce. Directed evolution of RNA enzyme. *Science*, **257**:635–641, 1992.
- [BM94] J. P. Bouchaud and X. Mézard. Self induced quenched disorder: a model for the glassy transition. *J. Phys. France*, **4**:1109, 1994.
- [BS93] T. Bäck and H.-P. Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*, **1**:1, 1993.
- [CA80] D. M. Ceperley and B. J. Alder. *Phys. Rev. Lett.*, **45**:566, 1980.
- [CA87] D. M. Ceperley and B. J. Alder. *Phys. Rev. B*, **36**:2092, 1987.
- [CFRS97] Jan Cupal, Christoph Flamm, Alexander Renner, and Peter F. Stadler. Density of states, metastable states, and saddle points. Exploring the energy landscape of an RNA molecule. In T. Gaasterland, P. Karp, K. Karplus, Ch. Ouzounis, Ch. Sander, and A. Valencia, editors, *Proceedings of the ISMB-97*, pages 88–91, Menlo Park, CA, 1997. AAAI Press.
- [CHS96] J. Cupal, I. L. Hofacker, and P. F. Stadler. Dynamic programming algorithm for the density of states of RNA secondary structures. In R. Hofstädt, T. Lengauer, M. Löffler, and D. Schomburg, editors, *Computer Science and Biology 96 (Proceedings of the German Conference on Bioinformatics)*, pages 184–186, Leipzig, Germany, 1996. Universität Leipzig.
- [Chv75] V. Chvátal. A combinatorial theorem in plane geometry. *J. Combin. Theory B*, **13**:395–398, 1975.
- [CN88] W-p. Chin and S. Ntafos. Optimum watchman routes. *Inf. Proc. Lett.*, **28**:39–44, 1988.

- [Dit96] F.-M. Dittes. Optimization on rugged landscapes: a new general purpose monte carlo approach. *Phys. Rev. Lett.*, 76:4651 – 4655, 1996.
- [EE86] W. Ebeling and A. Engel. Models of Evolutionary systems and their applications to optimization problems. *Syst. Anal. Model. Simul.*, 3:377, 1986.
- [EEF90a] W. Ebeling, A. Engel, and R. Feistel. *Physik der Evolutionsprozesse*. Akademie-Verlag, Berlin, 1990.
- [EEF90b] W. Ebeling, A. Engel, and R. Feistel. *Physik der Evolutionsprozesse*. Akademie-Verlag, berlin, 1990.
- [EF82] W. Ebeling and R. Feistel. *Physik der Selbstorganisation und Evolution*. Akademie-Verlag, Berlin, 1982.
- [EFR96] W. Ebeling, J. Freund, and K. Rateitschak. Entropy and extended memory in extended discrete chaotic dynamics. *Int. J. Bifurcation & Chaos*, 5:6, 1996.
- [Eig71] M. Eigen. The selforganization of matter and the evolution of biological macromolecules. *Naturwiss.*, 58:465, 1971.
- [EN92] W. Ebeling and G. Nicolis. *Chaos, Solitons & Fractals*, 2:635, 1992.
- [EN95] W. Ebeling and A. Neiman. Long range correlations between letters and sentences in texts. *Physica A*, 215:233, 1995.
- [ENP95] W. Ebeling, A. Neiman, and T. Pöschel. Dynamic entropies, long-range correlations and fluctuations in complex linear structures. In *Coherent Approach to Fluctuations*, Singapore, 1995. Hayashibara Forum 1995, World Scientific.
- [ENP96] W. Ebeling, A. Neiman, and T. Pöschel. *Coherent Approach to Fluctuations*. World Scientific, 1996.
- [EP94] W. Ebeling and T. Pöschel. Entropy and long-range correlations in literary english. *Europhys. Lett.*, 26:241, 1994.
- [EPA95] W. Ebeling, T. Pöschel, and K-F. Albrecht. Entropy, trans-information and word distributions of information-carrying sequences. *Int. J. Bifurc. & Chaos*, 5:241, 1995.

- [EPA96] W. Ebeling, T. Pöschel, and A. Neiman. Entropy and compressibility of symbol sequences. In J. Leao T. Toffoli, M. Biale, editor, *PhysComp96*, Cambridge, MA, 1996. New England Complex systems Institute.
- [ERS94] W. Ebeling, H. Rosé, and J. Schuchhardt. Evolutionary strategies for solving frustrated problems. In *Proceedings of the First World Congress of Computational Intelligence*, pages 79–81, Orlando, June 27-29 1994. IEEE.
- [ESG88] W. Ebeling and L. Schimansky-Geier. Stochastic processes in highdimensional systems and models of evolution. In P.E.V. McClintock F. Moss, editor, *Noise and Nonlinear Dynamical Systems*. Cambridge-University Press, 1988.
- [ESTCht] W. Ebeling, F. Schweitzer, B. Tilch, and V. Calenbuhr. Brownian motion models of animal mobility including energy depots. *J. Math. Bio.*, eingereicht.
- [Fei77] R. Feistel. Betrachtung der Realisierung stochastischer Prozesse aus automatentheoretischer Sicht. *Wiss. Z. W. Pieck Univ. Rostock*, **26**:663, 1977.
- [FERck] J. Freund, W. Ebeling, and K. Rateitschak. Self-similar sequences and universal scaling of dynamical entropies. *Phys. Rev. E*, im Druck.
- [Fey48] R.P. Feynman. Space-time approach to non-relativistic quantum mechanics. *Rev. Mod. Phys.*, **20**:267, 1948.
- [Fey72] R.P. Feynman. *Statistical Mechanics – A Set of Lectures*. Addison-Wesley, New York, 1972.
- [Fis30] R.A. Fisher. *The Genetical Theory of Natural Selection*. Oxford University Press, Oxford, 1930.
- [Fis78] S. Fisk. A short proof of chvatal’s watchman theorem. *J. Combin. Theory B*, 24, 1978.
- [Fre88] D.S. Freed. An index theorem for families of Fredholm operators parametrized by a group. *Topology*, **27**:279–300, 1988.
- [Fre95] J. Freund. *Dynamische Entropien und nichtlineare Prozesse mit langreichweitigen Korrelationen*. PhD thesis, Humboldt Universität zu Berlin, 1995.

- [Fri94] T. Fricke. *Neue Algorithmen zur Simulation von Zufallsprozessen*. Verlag der Augustinus Buchhandlung, Aachen, 1994.
- [FSBB<sup>+</sup>93] W. Fontana, P.F. Stadler, E.G. Bornberg-Bauer, T. Griesmacher, I. Hofacher, M. Tacker, P. Tarazona, E.D. Weinberger, and P. Schuster. RNA folding and combinatorial landscapes. *Phys. Rev. E*, 47(3):2083–2099, 1993.
- [GG73] M. Golubitsky and V. Guillemin. *Stable Mappings and their Singularities*. Graduate Texts in Mathematics 14. Springer Verlag, New York-Heidelberg-Berlin, 1973.
- [GGJ79] N. M. R. Garey, R. L. Graham, and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, San Francisco, 1979.
- [Gil76] D.T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chem. reactions. *J. Comp. Phys.*, **22**:403–434, 1976.
- [Gil78] D.T. Gillespie. Monte carlo simulation of random walks with residence time dependent transition probability rates. *J. of Comp. Phys.*, **28**:435–450, 1978.
- [GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: a guide to the theory of NP-completeness*. W. H. Freeman, San Francisco, 1979.
- [GM81] P.A. Griffiths and J.W. Morgan. *Rational homotopy theory and differential forms*. Birkhäuser, Boston, 1981.
- [Gol89a] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, 1989.
- [Gol89b] D.E. Goldberg. *Genetic Algorithms in search, optimization and machine learning*. Readin, MA: Addison-Wesley, 1989.
- [Hec95] T. Heckman. A unique application of genetic algorithms to machine control. 1995.
- [HES94] H. Herzel, W. Ebeling, and A.O. Schmitt. Entropies of biosequences: The role of repeats. *Phys. Rev. E*, 50:5061, 1994.

- [HFS<sup>+</sup>94] I. Hofacker, W. Fontana, P. Stadler, L. Bonhoeffer, M. Tacker, and P. Schuster. Fast folding and comparison of RNA secondary structures. *Monatshefte f. Chemie*, 125:167–188, 1994.
- [Hig95] P.G. Higgs. Thermodynamic properties of transfer RNA: A computational study. *J.Chem.Soc.Faraday Trans.*, 91:2531–2540, 1995.
- [HKP91] J. Hertz, A. Krogh, and R. Palmer. *Introduction to the theory of neural computation*. Addison–Wesley, Redwood City, 1991.
- [HM73] D. Husemoller and J. Milnor. *Symmetric Bilinear Forms*. Springer Verlag, Berlin, 1973.
- [HRW92] F.K. Hwang, D.S. Richards, and P. Winter. *The Steiner Tree Problem Annals of Discrete Mathematics*. North Holland, 1992.
- [HSG90] S. Harp, T. Samad, and A. Guha. Gesigning application–specific neural networks using the genetic algorithm. In *Advances in Neural Information Processing Systems 2*, San Mateo, 1990.
- [HT85] J. J. Hopfield and D. W. Tank. Neural computation of decisions in optimization problems. *Biol. Cyber.*, 52:141–152, 1985.
- [Ing89] L. Ingber. Very fast simulated annealing. *J. Math. Comp. Modelling*, 12:967, 1989.
- [Kap96] C. Kappler. Are evolutionary strategies improved by large mutations? In H.-M. Voigt et al., editors, *Proceedings of PPSNIV, Int. Conf. on Evolutionary Computation*, Berlin, 1996. Springer.
- [KGJV83] S. Kirkpatrick, C.D. Gelatt Jr., and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
- [LA87] P.J.M. Laarhoven and E.H.C. Aarts. *Simulated Annealing: Theory and Applications*. Reidel, Dordrecht, 1987.
- [LL87] L.D. Landau and E.M. Lifschitz. *Lehrbuch der theoretischen Physik*, volume 3. Akademie Verlag, Berlin, 1987.
- [Mil63] J. Milnor. *Morse theory*. Princeton Univ. Press, Princeton, 1963.

- [MRR<sup>+</sup>53] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1092, 1953.
- [NC91] S. Ntafos and W-p. Chin. Shortest watchman routes in simple polygons. *Discrete and Comp. Geom.*, 6:9, 1991.
- [NS88] J.D. Nulton and P. Salamon. Statistical mechanics of combinatorial optimization. *Phys. Rev. A*, 37(4):1351–1356, 1988.
- [Nta86] S. Ntafos. On gallery watchmen in grids. *Inf. Proc. Lett.*, 23:99–102, 1986.
- [Nta92] S. Ntafos. Watchman routes under limited visibility. *CGTA: Computational Geometry: Theory and Applications*, 1:149, 1992.
- [O'R87] J. O'Rourke. *Art Gallery Theorems*. Oxford Univ. Press, Oxford, 1987.
- [Pen92] C. K. Peng. *Nature*, 356:168, 1992.
- [PER95] T. Pöschel, W. Ebeling, and H. Rose. Guessing probabilities from small samples. *J. Stat. Phys.*, 80:1443, 1995.
- [REA96] H. Rosé, W. Ebeling, and T. Asselmeyer. The density of states - a measure of the difficulty of optimisation problems. In Voigt et al. [VERS96], pages 208–217.
- [Rec73a] I. Rechenberg. *Evolutionstrategie; Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog, Stuttgart-Bad Cannstatt, 1973.
- [Rec73b] I. Rechenberg. *Evolutionstrategien - Optimierung technischer Systeme nach Prinzipien der biologischen Information*. Friedrich Frommann Verlag (Günther Holzboog K.G.), Stuttgart - Bad Cannstatt, 1973.
- [REF96] K. Rateitschak, W. Ebeling, and J. Freund. Nonlinear dynamical model for texts. *Europhys. Lett.*, 35:401, 1996.
- [Rei]
- [RFE96] K. Rateitschak, J. Freund, and W. Ebeling. Entropy of sequences generated by nonlinear processes: the logistic map. In J. S. Shiner, editor, *Entropy and entropy generation*. Kluwer & Dordrecht, 1996.

- [Ros96a] H. Rosé. Von Boltzmann und Darwin zum Self Annealing – Strategien evolutionärer Optimierung. In J. Freund, editor, *Dynamik, Evolution, Strukturen*. Verlag Dr. Köster, Berlin, 1996.
- [Ros96b] D. Rosenkranz. Anwendung evolutionärer algorithmen auf quantenmechanische probleme. In J. Freund, editor, *Dynamik, Evolution, Strukturen: Nichtlineare Dynamik und Statistik komplexer Strukturen*. Verlag Dr. Köster, Berlin, 1996.
- [Rud95] O. Rudzick. Evolutionäre suche nach sequenzen mit kleinen autokorrelationen (labs): Bestimmung der entartung der autokorrelationfunktion. Preprint des BMBF-Projektes EVOALG, 1995.
- [Sch77a] H.-P. Schwefel. *Evolution and optimum seeking*. John Wiley & Sons Inc., 1977.
- [Sch77b] H.-P. Schwefel. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*, volume **26** of *Interdisciplinary system research*. Birkhäuser Verlag, Basel, 1977.
- [Sch81a] H.-P. Schwefel. *Numerical Optimization of Computer Models*. Wiley, New York, 1981.
- [Sch81b] H.-P. Schwefel. *Numerical Optimization of Computer Models*. John Wiley & Sons, Chichester, UK, 1981.
- [Sch94] A. O. Schmitt. *Structural Analysis of DNA Sequences*. PhD thesis, Humboldt Universität zu Berlin, 1994.
- [Sch95] H. P. Schwefel. *Evolution and optimum seeking*. Wiley-Interscience, New-York, 1995.
- [SERW96] F. Schweitzer, W. Ebeling, H. Rosé, and O. Weiss. Network optimization using evolutionary strategies. In Voigt et al. [VERS96], pages 940–949.
- [SG96] L. Schimansky-Geier. Beschreibung evolutionärer Algorithmen durch 2-Teilchenreaktionen. private communication, 1996.
- [SH87] H. Szu and R. Hartley. Fast simulated annealing. *Phys. Lett. A*, 122:157, 1987.
- [SHJ93] C. Svarer, L. Hanson, and Larsen J. On design and evaluation of tapped-delay neural network architectures. In *IEEE International Conference on Neural Networks*, San Francisco, 1993.

- [Ste94] I. Stewart. How many guards in the gallery. *Scient. Am.*, 270:88–92, 1994.
- [Str63] R.L. Stratonovich. *Topics in the theory of random noise*, volume 1. Gordon & Breach, New York, 1963.
- [Sym92] R. Symons. Small catalytic RNAs. *Ann.Rev.Biochem.*, 61:641–671, 1992.
- [VERS96] H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors. *Parallel Problem Solving from Nature - PPSN IV. Proceedings of the 4th International Conference on Evolutionary Computation*, volume 1141 of *Lecture Notes in Computer Science*. Springer, Berlin, 1996.
- [vK81] N.G. van Kampen. *Stochastic processes in physics and chemistry*. North-Holland Publishing Company, Amsterdam-New York-Oxford, 1981.
- [WK88] D. Whitley and J. Kauth. Genitor: A different genetic algorithm. In *Proceedings of the 1988 Rocky Mountain Conference on Artificial Intelligence*, pages 88–101. Denver, 1988.
- [WSS91] D. Whitley, T. Starkweather, and D. Shaner. The traveling salesman and sequence scheduling : Quality solutions using genetic edge recombination. In L. Davis, editor, *Handbook of genetic algorithms*, pages 350–371. Van Nostrand Reinhold, New York, 1991.
- [YL96] X. Yao and Y. Liu. Fast evolutionary programming. In L. J. Fogel, editor, *Proc. 5th Conf. Evol. Programming*, 1996.