Ein einheitliches Modell für Populationsstrukturen in Evolutionären Algorithmen

Dissertation
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
der Universität Dortmund
am Fachbereich Informatik

von
Joachim Sprave

Dortmund
1999

Tag der mündlichen Prüfung:
17. September 1999

Dekan:
Prof. Dr. B. Reusch

Gutachter:
Prof.-Dr.-Ing. H.-P. Schwefel
Prof.-Dr.-Ing. Ch. Lindemann

Danksagung

Ich danke Prof. Dr.-Ing. Schwefel für die gute Zusammenarbeit und das in mich gesetzte Vertrauen sowie Prof. Dr-Ing. Lindemann für sein Interesse an meiner Arbeit. Weiterhin danke ich den Kollegen am Lehrstuhl für Systemanalyse, die mich fachlich und menschlich durch die Zeit meiner Promotion begleitet haben, insbesondere der Kaffeeschnorrergang für willkommene Ablenkungen.

Mein besonderer Dank gilt Dr. Günter Rudolph für viele fachliche Diskussionen und motivierende Anregungen als Kollege und als Freund, sowie Mike Preuß für seine Korrekturen und Anmerkungen zu dieser Arbeit.

Allen Freunden, Bekannten und Verwandten, insbesondere meinen Eltern, danke ich für das Verständnis, das sie mir in den letzten Monaten wissentlich oder unwissentlich entgegengebracht haben.

Und Moni danke ich einfach für alles.

Dortmund, Juni 1999

Joachim Sprave

Für Moni.

Inhaltsverzeichnis

1 Einleitung **5**

2 Evolutionäre Algorithmen **7**

2.1 Problemstellung 7

2.2 Schwierige Optimierungsaufgaben 9

2.3 Evolutionäre Algorithmen 10

2.3.1 Das gemeinsame Modell 10

2.3.2 Genetische Algorithmen 12

2.3.3 Evolutionsstrategien 18

2.3.4 Evolutionäres Programmieren 22

2.3.5 Varianten der Basisalgorithmen 23

2.3.6 ES+GA+EP+... = Evolutionäre Algorithmen 23

3 Räumliche Strukturen in parallelen EA **25**

3.1 Historie der parallelen EA 25

3.2 Klassifizierung nach Gorges-Schleuter 26

3.2.1 Ursprüngliche Klassifizierung 26

3.2.2 Verfeinerung der Klassifizierung 26

3.3 Räumliche strukturierte Populationsmodelle 28
3.3.1 Panmiktische Teilpopulationen 28
3.3.2 Nachbarschaftsmodelle 29
3.3.3 Gemischte parallele Populationsmodelle 30
3.4 Analytische Untersuchungen von Populationsstrukturen 30

4 Vereinheitlichte Modellierung von Populationsstrukturen 33
4.1 Motivation 33
4.2 Hypergraphen 33
4.2.1 k-Schnitte von Hypergraphen 35
4.2.2 Repräsentationen von Hypergraphen 36
4.2.3 Modellierung von Populationsstrukturen mittels Nachbarschaftsgraphen 38
4.2.4 Modellierung von Populationsstrukturen mittels Hypergraphen 39
4.3 Hypergraphmodellierung klassischer Populationsmodelle 42
4.3.1 Panmixie 42
4.3.2 Multipopulationsmodelle 42
4.3.3 Feinkörnige Modelle 44
4.3.4 Modellierung von Nachkommenüberschuß 46
4.4 Modellierung von Isolationszeiten 47
4.5 Vom Modell zum Programm 49

5 Selektion in Populationsstrukturen 53
5.1 Einleitung 53
5.2 Panmiktische Selektionsoperatoren 54

5.2.1 Proportionale Selektion 54
5.2.2 Rangselektion 55
5.2.3 Turnierselektion 56
5.2.4 $(\mu \overset{+}{,} \lambda)$-Selektion 57
5.3 Lokale Selektion 58
5.3.1 Lokale proportionale Selektion 58
5.3.2 Lokale Rangselektion 59
5.3.3 Lokale Turnierselektion 59
5.3.4 Lokale (μ, λ)-Selektion 59
5.3.5 Spezielle lokale Selektionsoperatoren 61

6 Takeover-Verhalten von Populationsmodellen 63
6.1 Der Begriff der Takeover-Zeit 63
6.2 Probabilistischer Durchmesser 69
6.3 Derandomisierung von Turnierselektion 75

7 Experimenteller Vergleich von Populationstrukturen 79
7.1 Ziele der Experimente 79
7.2 Populationsstrukturen und das Kugelmodell 81
7.3 Multimodale Zielfunktionen 84
7.4 Bewertung der Experimente 91

8 Weitere Konzepte 95
8.1 Restriktionsbehandlung für rangbasierte Algorithmen 95
8.2 Inhomogene Parallelisierung 102
8.3 Lokale adaptive Akzeptanzschwellen 102

9 Das PACE-Populationsmodell und seine Anwendung 111

9.1 Das PACE-Populationsmodell 111

9.2 Design optischer Mehrschichtsysteme 113

9.2.1 Problemstellung 113

9.2.2 Interne Repräsentation 115

9.2.3 Populationsstruktur und Parallelisierung 116

9.2.4 Ergebnisse 116

10 Zusammenfassung und Ausblick 119

Kapitel 1

Einleitung

Es gibt heute keine Konferenz im Bereich Evolutionärer Algorithmen (EA) mehr, auf der nicht mindestens ein Beitrag die Parallelisierung dieser Verfahren zum Thema hat. Waren es in der 80er Jahren im wesentlichen neue oder für neu gehaltene Konzepte von Parallelität sowie neue Naturanalogien Motivation der gewählten Kommunikationsschemata, so stehen nun eher Anwendungen im Vordergrund, die aus rein pragmatischen Gründen auf bewährte Parallelisierungsansätze zurückgreifen. Dabei zeichnet sich jedoch weder eine Kanonisierung der Terminologie noch der Theorie, soweit überhaupt vorhanden, ab. Dies führt dazu, daß ähnliche Ansätze unter völlig verschiedenen Namen mehrfach entdeckt werden, oder daß konzeptionell unterschiedliche Varianten mit nahezu gleichlautenden Bezeichnungen versehen werden.

Die uneinheitliche Terminologie kann nur zum Teil dadurch erklärt werden, daß sich Fachbegriffe bei einem relativ neuen Forschungsgebiet erst einmal etablieren müssen. Einen wesentlichen Einfluß hat nämlich auch die Tatsache, daß das Gebiet als solches nur schwer abzugrenzen ist, und, schlimmer noch, daß noch nicht einmal Einigkeit über den Kern dessen besteht, was mit dem Begriff Evolutionäre Algorithmen bezeichnet wird.

Historisch gesehen gibt es zwei Extreme. Auf der einen Seite stand die Idee, einfache Computermodelle zur Simulation der natürlichen Evolution zu entwickeln. Das vorrangige Ziel dabei war es, anhand einfacher Modelle mehr über das natürliche Vorbild zu lernen. Auf der anderen Seite gab es schon früh Ansätze, aus den Prinzipien der Evolution Strategien zur Optimierung technischer Systeme abzuleiten. Vor allem dadurch, daß John Holland eigentlich ersteres beabsichtigte, seine Ideen aber als Ansätze zu Optimierverfahren interpretiert wurden, wurden beinahe schon fahrlässig Begriffe aus der Populationbiologie und der Vererbungslehre sowie der Numerik, Kodierungstheorie und dem Bereich mathematischer Optimierung miteinander vermischt.

Bei den parallelen EA setzt sich diese Entwicklung fort. Häufig werden Begriffe aus der parallelen Datenverarbeitung und der Biologie im gleichen Kontext verwendet. Als besonders schwierig stellt sich daher die Bewertung einer vorgenommenen Parallelisierung dar. Für die zwei wesentlichen Eigenschaften einer parallelen Variante eines EA, nämlich die lokale Selektion und die räumliche Struktur der Population, fehlen formale Gerüste für eine einheitliche, abstrakte Darstellung.

Kapitel 2

Evolutionäre Algorithmen

Die Nachahmung von Prinzipien der Natur hat viele Erfindungen der Menschheit bewirkt oder beeinflußt. Während früher nur offensichtlich nützliche Prinzipien aus der Natur wie etwa der Vogelflug zum Vorbild für technische Entwicklungen genommen wurden, wird heute gezielt nach nachahmenswerten Lösungen in der Tier- und Pflanzenwert gesucht. Zu diesem *Bionik* genannten Forschungsgebiet gehört auch die Idee, daß nicht nur die Ergebnisse der biologischen Evolution Wege zu technischen Lösungen aufzeigen können, sondern auch die Entwicklungs- und Anpassungsprozesse, die zu diesen Lösungen geführt haben. Aus diesem Gedanken heraus entstanden bereits in den 60er Jahren unabhängig voneinander mehrere Verfahren, die die Prinzipien der natürlichen Evolution als Modell für Adaptions- oder Optimierungsstrategien verwendeten. Heute faßt man sie unter dem Oberbegriff *Evolutionäre Algorithmen* zusammen. Diese werden wiederum neben den Künstlichen Neuronalen Netzen und der Fuzzy Logic als Teilgebiet der *Computational Intelligence* betrachtet. Die folgenden Absätze geben einen kurzen Überblick über wesentliche Aspekte Evolutionärer Algorithmen. Für ein umfassendes Studium dieses Themengebiets sei der Leser auf das *Handbook of Evolutionary Computation* (Bäck, Fogel und Michalewicz 1997) verwiesen.

2.1 Problemstellung

Die wesentlichen Anwendungsgebiete Evolutionärer Algorithmen liegen im Bereich der reellwertigen Parameteroptimierung, der pseudo-Booleschen Optimierung und und der Strukturoptimierung.

Als Optimierung[1] bezeichnet man die Verbesserung eines gegebenen Systems anhand festgelegter Gütekriterien. Dabei untersucht man üblicherweise vereinfachende Modelle dieser Systeme, zum einen, um die Komplexität der Aufgabe zu reduzieren, zum anderen, weil es oft nicht möglich, zu teuer oder zu gefährlich ist, reale Systeme zu beeinflussen. In vielen Fällen entziehen sich die Systeme oder deren Simulationen einer mathematischen Analyse. Man ist dann darauf angewiesen, die Reaktion des Systems auf Veränderungen der beeinflußbaren Parameter zu messen. Man ordnet so dem System eine Qualitätsfunktion zu, die implizit oder in Form eines mathematischen Modells gegeben sein kann. Diese bezeichnet man als *Zielfunktion* mit

$$f : M \to \mathrm{I\!R},\ M \subseteq \mathrm{I\!R}^n, \tag{2.1}$$

wobei üblicherweise f so formuliert wird, daß ein Punkt $\vec{x} \in M$ um so besser ist, je kleiner $f(\vec{x})$ ist (Minimierung). Ein Punkt $\vec{x}^*$ im Parameterraum heißt dann *globaler Optimumpunkt* und sein Zielfunktionswert $f(\vec{x}^*)$ *globales Optimum*, wenn gilt:

$$\forall \vec{x} \in M : f(\vec{x}) \geq f(\vec{x}^*). \tag{2.2}$$

Verlangt man die Erfüllung der Ungleichung 2.2 nur für alle $\vec{x}$ in einer beliebig kleinen ε-Umgebung von $\vec{x}^*$, so spricht man von einem *lokalen Optimumpunkt* bzw. einem *lokalen Optimum*. Das globale Optimum ist also immer auch ein lokales. Existieren zu einer Zielfunktion mehrere, voneinander verschiedene lokale Optima, so spricht man auch von einer *multimodalen* Zielfunktion.

Der Lösungsraum M kann durch Gleichheits- oder Ungleichheitsrestriktionen beschränkt sein:

$$\begin{aligned} M &:= \{\vec{x} \in \mathrm{I\!R}^n \mid \forall k = 1 \ldots q : g_k(\vec{x}) \geq 0\} \\ &\cap \{\vec{x} \in \mathrm{I\!R}^n \mid \forall k = 1 \ldots p : h_k(\vec{x}) = 0\}. \end{aligned} \tag{2.3}$$

Ungleichheitsrestriktionen in Form untereinander unabhängiger Ober- und Untergrenzen der Parameter (Bounds) werden oft nicht als Restriktionen im eigentlichen Sinne betrachtet.

[1] Streng genommen trifft dieser Begriff nur auf die Suche nach der besten Lösung zu. Der präzisere Begriff der *Meliorisierung* für Verbesserungsverfahren ist aber eher unüblich.

Abzählbare Suchräume sind in der obigen Definition als endliche Teilmengen von $\mathbb{R}^n$ natürlich enthalten. Dennoch wird gelegentlich zwischen kontinuierlicher, diskreter und gemischt-ganzzahliger Optimierung unterschieden.

2.2 Schwierige Optimierungsaufgaben

Wenn die Zielfunktion explizit als mathematischer Ausdruck formulierbar ist, kann man versuchen, das Optimum analytisch zu bestimmen. Aus der Algebra wissen wir aber, daß dies schon bei so einfachen Funktionen wie Polynomen höheren Grades nicht gelingt. Doch immerhin kann man die Kenntnis gewisser Eigenschaften wie Stetigkeit oder Differenzierbarkeit nutzen, um ein numerisches Verfahren zur Lösung eines solchen Problems zu entwickeln. Geht ein Verfahren davon aus, daß die zu bearbeitende Zielfunktion gewisse Annahmen erfüllt, so besitzt es ein *inneres Modell* der Zielfunktion. Wendet man nun dieses Verfahren auf ein Problem an, das nicht die Charakteristika des inneren Modells aufweist, so wird es in aller Regel versagen. Das Newton-Raphson-Verfahren etwa startet an einem Punkt im Suchraum und ermittelt anhand der partiellen Ableitungen in diesem Punkt gewissermaßen einen Vorschlag für die Lage des Optimums. Ist die Zielfunktion quadratisch, entspricht sie dem inneren Modell des Verfahrens und das Optimum wird in einem Schritt gefunden. Besitzt die Zielfunktion aber mehrere lokale Optima, so hängt es vom Startpunkt ab, ob zumindest ein lokales Optimum gefunden wird oder das Verfahren sogar divergiert.

In der Praxis ist die Zielfunktion häufig nur implizit durch ein Simulationsmodell oder durch Messungen an einem realen System gegeben. Dann können nicht einmal Eigenschaften wie Stetigkeit oder Kontinuität zugesagt werden. Die Zielfunktion ist dann eine Black-Box mit einer Vielzahl von Einstellknöpfen zur Einstellung der Parameter und einer Skala zur Anzeige der Güte der aktuellen Einstellung. Dies sind die wirklich schwierigen Optimieraufgaben: Verfahren, die ein solches Problem lösen sollen, dürfen kein festes inneres Modell besitzen, sondern müssen in der Lage sein, zu einer gegebenen Zielfunktion das innere Modell zu erlernen.

Im Falle der Optimierung anhand von Meßwerten treten noch weitere Schwierigkeiten auf. Durch Meßfehler können scheinbar sehr gute Lösungen entstehen, die die Suche in eine völlig falsche Richtung lenken. In dieser Situation ist es erforderlich, daß ein Verfahren die Fähigkeit besitzt zu vergessen. Weiterhin ist die

Lage des Optimums eines realen Systems nicht notwendigerweise zeitinvariant. Driftet ein einmal gefundenes Optimum langsam weg, so muß es gewissermaßen verfolgt werden. Verändert sich die Topologie derart, daß laufend neue Optima entstehen und alte verschwinden, sollte ein Verfahren in der Lage sein, neue Gebiete im Suchraum zu erforschen, wenn lokal kein Fortschritt mehr zu erzielen ist.

2.3 Evolutionäre Algorithmen

2.3.1 Das gemeinsame Modell

Evolutionäre Algorithmen (EA) verwenden die folgenden Analogien: Eine potentielle Lösung eines Optimierproblems, also ein Punkt im Suchraum, wird als Individuum in einer künstlichen Umwelt angesehen, der Grad der Anpassung an diese Umwelt wird durch die Zielfunktion bestimmt. Die Koordinaten im Suchraum (bzw. eine irgendwie gewählte Repräsentation der Koordinaten) werden meist als Genom eines Indivduums betrachtet.

Faßt man nun noch mehrere Individuen zu einer *Population* zusammen, so kann man weitere Mechanismen modellieren:

- Reproduktion:
  Aus den gegebenen Punkten im Suchraum, der aktuellen *Generation*, können analog zur Fortpflanzung in der Natur neue Punkte generiert werden.

- Mutation:
  Bei der Reproduktion sorgen kleine Veränderungen der genetischen Informationen für die notwendige Variation. Mutationen sind die Tastschritte der Natur, wobei häufig kleine, nur sehr selten aber große Änderungen auftreten.

- Selektion:
  Seit Charles Darwin geht man davon aus, daß in der Natur besser angepaßte Individuen größere Chancen haben, zu überleben und sich fortzupflanzen.

Zur Übertragung dieser Beobachtung auf die Optimierung benötigt man also eine Bewertung. Diese ist durch die Zielfunktion gegeben, der Funktionwert dient als *Fitneß* in einer künstlichen Umwelt.

- Rekombination:
  Auch die geschlechtliche Vererbung wird in EA nachgeahmt. Hierzu werden aus zwei oder mehr Elternindividuen durch zufälliges Kopieren von Bruchstücken der Elterngenome Nachkommen erzeugt.

Da insbesondere Mutation und Rekombination in der Natur zufällig und ungerichtet zu sein scheinen, werden diese bei den meisten EA mit Hilfe von Zufallsprozessen modelliert.

Auf dem folgenden Basisalgorithmus basieren alle EA mehr oder weniger:

$t := 0$

Initialisiere P^t

Bewerte P^t

Solange nicht Abbruch

S$_M$: P'^t :=Partnerselektion(P^t)
R: P''^t :=Rekombination(P'^t)
M: P'''^t :=Mutation(P''^t)
Auswertung von P'''^t
S$_E$: P^{t+1} :=Umweltselektion$(P'''^t \cup P^t)$
t := t + 1.

Dabei steht P^t für die Elterngeneration, P'''^t für die Nachkommengeneration zum Zeitpunkt t. Die Verwendung von zwei Selektionszeitpunkten ist nicht unbedingt erforderlich, da sie in der zyklischen Abfolge direkt hintereinander ausgeführt werden. Um aber verschiedene Varianten, die entweder Selektion als Partnerwahl (mating selection) oder aber als Umweltselektion (environmental selection) modellieren, in einem Ablaufschema zu erfassen, verwenden neuere Definitionen Evolutionärer Algorithmen dieses erweiterte Schema (Schwefel und Kursawe 1998).

Obwohl das algorithmische Grundgerüst gleich ist, soll hier zunächst auf die weitgehend unabhängig voneinander entstandenen, historischen Basisalgorithmen eingegangen werden.

2.3.2 Genetische Algorithmen

John Holland (1975) entwickelte Ende der 60er Jahre ein naturanaloges Adaptionsverfahren, dem er den Namen *Genetischer Algorithmus* gab. Die Problemparameter werden dabei analog zu den DNA-Sequenzen als Zeichenketten über einem endlichen Alphabet, in der Praxis fast ausschließlich über der Menge $\mathbb{B} := \{0, 1\}$, kodiert. Zur Behandlung reellwertiger Funktionen wird üblicherweise der Bit-String eines Individuums in folgender Weise in eine Festkommadarstellung abgebildet:

$$x_i \quad := \quad \frac{\sum_{\nu=0}^{w-1} (b_{w \cdot i+\nu} \cdot 2^{\nu}) - 2^{w-1}}{10^K}. \tag{2.4}$$

Dabei ist w die Anzahl der Bits, die einen Parameter kodieren, $b_0 \ldots b_{w-1}$ der Bit-String, K die Anzahl der Nachkommastellen und somit $-2^{w-1}/10^K$ der kleinste und $(2^{w-1} - 1)/10^K$ der größte darstellbare Wert. Vor jeder Zielfunktionsauswertung wird also noch eine sogenannte *Mapping*-Funktion m, $m : \mathbb{B}^{w \cdot n} \to \mathbb{R}^n$, ausgewertet. Wir erhalten eine resultierende Fitneß-Funktion

$$F : \mathbb{B}^{w \cdot n} \to \mathbb{R}, \; F(\vec{b}) = f(m(\vec{b})). \tag{2.5}$$

Schemata

Eine der Annahmen Hollands war es, daß durch eine Population von in Bit-Strings kodierten Individuen nicht nur die vorhandenen Strings, sondern alle Kombinationsmöglichkeiten aus diesen repräsentiert werden, sobald man Rekombination zuläßt. Als Erklärungsversuch dient das Schema-Theorem, welches besagt, daß sich überdurchschnittlich gute Bitmuster exponentiell in einer Population ausbreiten. Ein Bitmuster wird dann auch als *Schema* bezeichnet und läßt sich über dem Alphabet $\{0, 1, \#\}$ kodieren. Das # fungiert dabei als Joker, für den eine 0 oder eine 1 eingesetzt werden kann. Die Länge eines Schemas ist die Länge des längsten Teilstrings, der weder mit einem Jokerzeichen beginnt noch mit einem solchen endet. Tabelle 2.3.2 verdeutlicht diese Begriffe anhand von konkreten Beispielen.

Schema	Passende Strings	Länge
11#0	1100, 1110	4
#1#0	0100, 0110, 1100, 1101	3
0##0	0000, 0010, 0100, 0110	4
#10#	0100, 0101, 1100, 1101	2
####	0000 ... 1111	0

Tabelle 2.3.2: Beispiele für Schemata.

Ablauf

Während bei Holland immer nur einzelne Individuen einer Population ersetzt wurden, hat sich schließlich eine feste Generationentaktung durchgesetzt, die in der obigen Notation den Ablauf **S–R–M** aufweist. Die meisten heute verwendeten Varianten gehen auf Goldberg (1989) zurück, wie auch die nachfolgend beschriebenen genetischen Operatoren.

Selektion

Die Modellierung der Selektion in GA beruht auf der Annahme, daß in der Natur die zu erwartende Anzahl der Nachkommen in irgendeiner Weise proportional zur Fitneß ist, die um so größer ist, je besser ein Individuum angepaßt ist. Daher werden Optimierprobleme bei klassischen GA als Maximierungsproblem formuliert.

Aus einer aktuellen Generation wird durch sogenannte *Roulette-Wheel-Selection* eine Elterngeneration gleicher Größe ermittelt. Die Vorstellung dabei ist, daß jedem Individuum eine von der Fitneß abhängige Anzahl von Slots auf einem Roulette-Rad zugeordnet werden. Der Anteil eines Individuums an den insgesamt verfügbaren Slots wird *relative Fitneß* genannt. Nun wird das Rad sooft gedreht, wie Plätze in der Population vorhanden sind. Jedesmal, wenn das Rad zum Stillstand kommt, wird das Individuum, das den Slot an der Haltemarkierung besetzt, in die Elterngeneration übernommen.

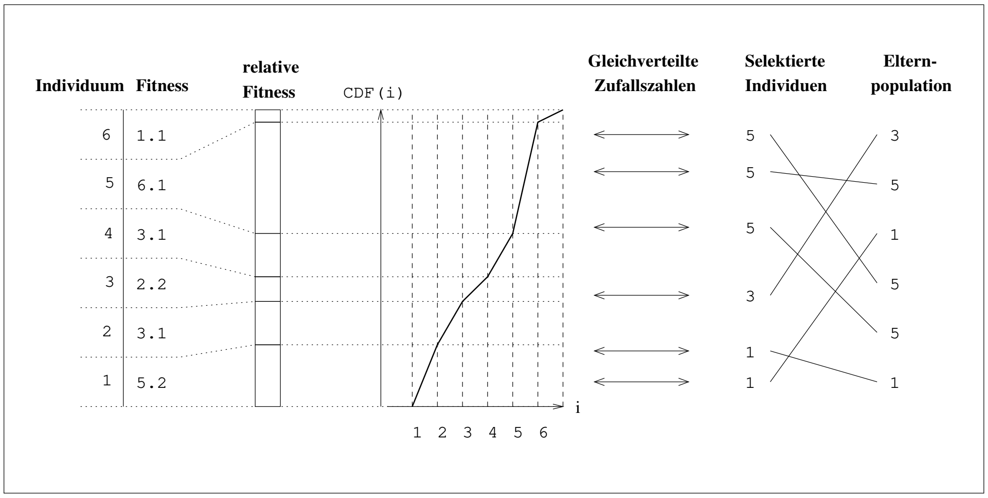

Abbildung 2.1: Proportionale Selektion (schematisch).

Die relative Fitneß entsteht rechnerisch, indem man jeden Fitneßwert innerhalb einer Population durch die Summe aller Fitneßwerte teilt. Man teilt dann das Intervall $[0\ldots1]$ so, daß zu jedem Individuum ein Intervall mit Länge seiner relativen Fitneß existiert. Zur Selektion ermittelt man nun Zufallszahlen gleichverteilt zwischen 0 und 1 und selektiert jeweils das Individuum, in dessen Intervall die aktuelle Zufallszahl liegt.

Zur Bestimmung der Intervallgrenzen verwendet man die sogenannte kumulative Dichtefunktion (CDF), die zu jedem Individuum die Summe der relativen Fitneßwerte der Vorgängerindividuen liefert. Abb. 2.1 stellt den Zusammenhang zwischen relativer Fitneß und CDF schematisch dar.

Sei nun $|P|$ die Anzahl der Individuen in der Population, P^t die aktuelle Generation von Individuen $p_i, i = 1\ldots|P|$ und $G_{[0\ldots1]}$ eine auf dem Intervall $[0\ldots1]$ gleichverteilte reelle Zufallsvariable, dann wird die Elterngeneration $P'^t = \{p'_1 \ldots p'_{|P|}\}$ wie folgt bestimmt:

$$\begin{aligned} p'_i &:= p_k,\ i = 1,\ldots,|P| && (2.6)\\ k &= \min\left\{j \in \{1,\ldots,|P|\} \mid \mathrm{CDF}(j) \geq G_{[0\ldots1]}\right\}, && (2.7) \end{aligned}$$

mit

$$\mathrm{CDF}(j) \quad := \quad \frac{\sum_{\nu=1}^{j} F(p_\nu)}{\sum_{\kappa=1}^{|P|} F(p_\kappa)}\,. \tag{2.8}$$

Man beachte, daß eine negative relative Fitneß keinen Sinn ergibt. Daraus folgt, daß die Zielfunktion immer positive Werte liefern muß, damit proportionale Selektion anwendbar ist. Weiterhin sind häufig gegen Ende einer Optimierung mit Genetischen Algorithmen alle Fitneßwerte weit größer als Null, so daß die Individuen annähernd gleiche relative Fitneß aufweisen und somit Selektion kaum noch stattfindet. Abhilfe schaffen Window-Techniken, mit Hilfe derer der Bezugswert für die relative Fitneß im Laufe des Verfahrens immer weiter hochgeschraubt wird, und Methoden zur Skalierung der Zielfunktion.

Rekombination

Die in der Elterngeneration P'_t befindlichen Individuen werden paarweise zu jeweils zwei Nachkommen rekombiniert, so daß wiederum die Populationsgröße unverändert bleibt. Zu je zwei Eltern wird zufällig eine Bruchstelle im Genom ermittelt. An dieser Stelle werden die hinteren Bruchstücke der beiden Elterngenome abgetrennt und an das vordere Bruchstück des jeweils anderen Genoms angehängt (Abb. 2.2). Wegen der Verwendung nur einer Bruchstelle heißt dieses Verfahren auch *1-Point-Crossover* und ist neben dem 2-Point-Crossover das wohl am häufigsten verwendete Rekombinationsschema.

Die Wahl von nur ein oder zwei Bruchstellen ist durch das Schematheorem motiviert. Kurze Schemata, die eine hohe Fitneß bewirken, sollen mit hoher Wahrscheinlichkeit nicht aufgetrennt werden. Dennoch werden in letzter Zeit mit *Uniform-Crossover*, bei dem jedes Bit zufällig mit gleicher Wahrscheinlichkeit von einem der Eltern ausgewählt wird, gute Erfolge erzielt.

Mutation

Zur Nachahmung der biologischen Mutation wird jedes Bit eines erzeugten Nachkommen mit einer sehr geringen Wahrscheinlichkeit, der *Mutationsrate* invertiert. Bezogen auf ein in der Kodierungstheorie gängiges Maß, den Hamming-Abstand,

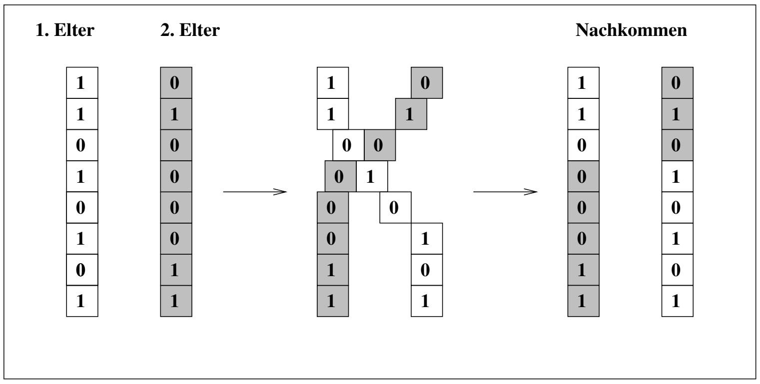

Abbildung 2.2: One-Point-Crossover

erfüllt dieses Vorgehen die Forderung nach in der Regel kleinen Änderungen: Im Alphabet $\mathbb{B}^N$ ist der Hamming-Abstand gerade die Anzahl der unterschiedlichen Bits zweier Wörter. Der Mutation wird ohnehin in GA keine große Bedeutung zugemessen, im klassischen Fall liegen die Mutationsraten etwa bei 10^{-3}. Es wird davon ausgegangen, daß bei genügend großer Population bereits eine ausreichende Anzahl guter Schemata initial vorhanden ist und nahezu allein durch Rekombination ein Optimum gefunden werden kann.

Probleme der Bit-Kodierung

Sind die Bit-Strings Kodierungen reeller Parameter, so ist ein kleiner Hamming-Abstand zwischen zwei Parameterkodierungen keine Garantie dafür, daß auch die dekodierten Werte im Parameterraum nahe beieinanderliegen. Zwei in jeweils 4-Bit kodierte ganze Zahlen mit einem Hamming-Abstand von 1 können einen euklidischen Abstand von 1, 2, 4 oder 8 aufweisen, die Wirkung einer 1-Bit-Änderung ist vom Stellenwert eines invertierten Bits innerhalb eines Parameters abhängig. Umgekehrt bedeutet dies, daß es für einen GA schwierig sein kann, eine im Parameterraum leicht zu erreichende Verbesserung zu erzielen. Nehmen wir als Beispiel eine sehr einfache, eindimensionale Funktion

$$f \quad : \quad \mathbb{N} \to \mathbb{R},\ f(x) = 100 - (8 - x)^2, \tag{2.9}$$

Generation	Bester Wert	
	dezimal	binär
1	2	0010
2	3	0011
3	3	0011
4	6	0110
5	7	0111

Tabelle 2.3.2: Mögliche Entwicklung des besten Individuums in einem GA mit der Zielfunktion aus Gl. 2.9.

mit einer binären Kodierung im üblichen Stellenwertsystem:

$$x_i = \sum_{\nu=0}^{3} b_\nu \cdot 2^\nu \; . \tag{2.10}$$

Offensichtlich nimmt die Funktion ihr globales Maximum von 100 für $x = 8$ an. Hier kann nun der Fall eintreten, daß alle Individuen einer Population sich der Optimumstelle aus dem unteren Zahlenbereich heraus nähern. Der Wert des besten Individuums der jeweils aktuellen Generation könnte sich dann so entwickeln wie in Tabelle 2.3.2.

Um allein durch Mutation von $7 = 0111_2$ auf $8 = 1000_2$ zu kommen, müssen am Ende alle Bits gleichzeitig invertiert werden. Bei längeren Kodierungen als in diesem Beispiel ist dies sehr unwahrscheinlich.

Dies wäre kein Problem, wenn diese Situation relativ selten aufträte. Aber gerade in der Nähe eines Optimums unterscheiden sich die Individuen einer Population kaum noch. Im vorliegenden Fall würden mit hoher Wahrscheinlichkeit alle Strings, die auf die Schemata `#0##` und `##0#` passen, durch die Selektion herausfallen, so daß auch durch Rekombination der Sprung zum Optimum nicht zu erwarten ist. Man nennt diese Schwierigkeit auch *Hamming-Cliff*.

2.3.3 Evolutionsstrategien

Obwohl die in den 60er Jahren von Rechenberg (1965) und Schwefel (1965) entwickelten Evolutionsstrategien zunächst zur diskreten, experimentellen Optimierung, beispielsweise von Klockgether und Schwefel (1970), eingesetzt wurden, entwickelten sie sich vor allem zu einem Instrument zur reellwertigen Parameteroptimierung (Rechenberg 1973; Schwefel 1977).

Zweigliedrige Evolutionsstrategie

In der ursprünglichen Fassung gab es noch nicht einmal eine echte Population: Bei der zweigliedrigen ES wird mit einem Individuum gestartet, das einen Nachkommen erzeugt. Das bessere der beiden überlebt und wird zum Elternindividuum der nächsten Generation. Als einziger Mechanismus zur Variation wird die Mutation eingesetzt: Zur Erzeugung des Nachkommen wird zu jedem Parameter ein Zufallswert einer Normalverteilung mit Mittelwert 0 und Standardabweichung σ addiert. Damit ist sichergestellt, daß der Nachkomme mit hoher Wahrscheinlichkeit einen geringen euklidischen Abstand von seinem Vorfahren hat, aber dennoch mit positiver Wahrscheinlichkeit jeden Punkt im Suchraum erreichen kann.

Die Standardabweichung der Normalverteilung kann als mittlere Schrittweite angesehen werden und kann entweder fest eingestellt sein oder während des Verlaufs der Optimierung adaptiert werden, etwa nach der verblüffend einfachen, empirisch hergeleiteten $1/5$-Erfolgsregel von Rechenberg (1973), die sinngemäß lautet:

Das Verhältnis erfolgreicher Mutationen zu allen Mutationen sollte etwa $1/5$ betragen. Ist es größer, muß die Schrittweite vergrößert werden; ist es kleiner, so muß die Schrittweite verringert werden.

Mehrgliedrige Evolutionsstrategien

Die Einführung von größerer Elternanzahlen (Rechenberg 1973) und größerer Nachkommenanzahlen (Schwefel 1977) führte zur mehrgliedrigen ES, die dabei auch um Rekombinationsmechanismen erweitert wurde. Eine Elternpopulation von μ Individuen erzeugt durch Rekombination und Mutation λ Nachkommen. Eltern und Nachkommen werden einem gemeinsamen, deterministischen Selektionsprozeß unterworfen, den die besten μ Individuen der $\mu + \lambda$ zur Auswahl stehenden überleben. Diese Variante der ES heißt auch $(\mu + \lambda)$-ES oder *Plus-Strategie*.

Dadurch, daß gute Individuen beliebig lange leben können, fehlt der Plus-Strategie die oben beschriebene Fähigkeit zu vergessen. Daher wurde als Alternative zur Plus-Strategie die sogenannte *Komma-Strategie* (Schwefel 1977) entwickelt. Bei der (μ, λ)-ES werden aus μ Eltern $\lambda > \mu$ Nachkommen erzeugt. Die besten μ Nachkommen überleben und werden zur nächsten Elterngeneration.

Schrittweitensteuerung

Mit Hilfe der Schrittweitensteuerung passen sich ES jeweils lokal an die Zielfunktionstopologie an. Anstelle einer heuristischen Anpassung gehen moderne ES einen anderen Weg: Jedes Individuum bekommt zusätzlich zu seinen Problemparametern noch eine eigene Schrittweite als Strategieparameter, der ebenfalls vererbt und evolutionären Prozessen unterzogen wird.

Allerdings kann eine gemeinsame Schrittweite für alle Parameter hier nur ein Kompromiß sein, wenn die Zielfunktion stark gegenüber den Achsen verzerrt ist oder Schluchten und Grate aufweist. Wenn man mit einem derartigen Verhalten rechnet oder überhaupt keine Information über den Grad der Zerklüftung der Zielfunktion hat, ist es sinnvoll, für jede Objektvariable eine eigene Schrittweite mitzuführen.

Rekombination in ES

Während die proportionale Selektion bei Genetischen Algorithmen nicht nur festlegt, ob sich ein Individuum fortpflanzt, sondern auch wie viele Nachkommen es haben soll, sind Selektion und Reproduktion bei den ES voneinander getrennt. Alle selektierten Individuen werden mit gleicher Wahrscheinlichkeit zur Rekombination herangezogen. Üblicherweise werden λ mal jeweils zwei oder mehr Individuen zufällig bestimmt, um durch Rekombination einen Nachkommen zu erzeugen, der anschließend noch mutiert wird.

Die reellwertige Kodierung der Parameter führt zu zwei naheliegenden Rekombinationsvarianten. Bei der intermediären Rekombination wird an jeder Stelle im

Parametersatz der arithmetische Mittelwert aus den beiden Elternparametern gebildet. Dagegen wird bei der sogenannten diskreten Rekombination jeder Parameter mit gleicher Wahrscheinlichkeit von einem der Eltern unverändert übernommen. Letzteres entspricht auf der Repräsentationsebene dem Uniform-Crossover-Operator der Genetischen Algorithmen. Ein Beispiel für diskrete Rekombination auf den Objektvariablen und intermediäre Rekombination auf den Schrittweiten im Falle von zwei Eltern liefert Abb. 2.3.

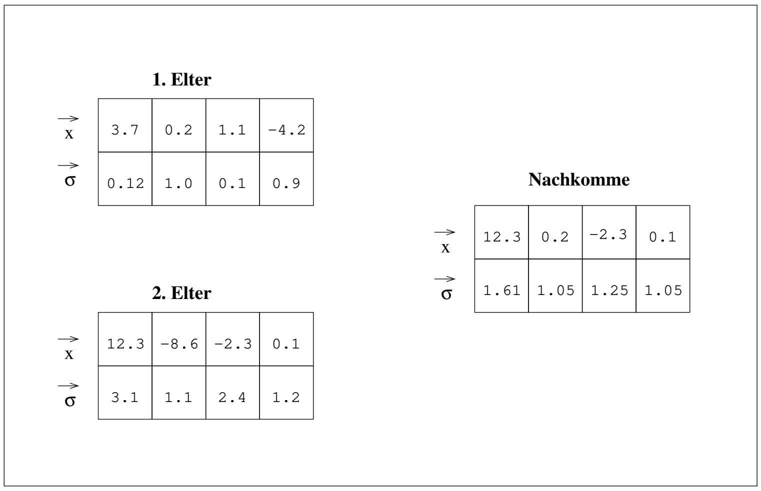

Abbildung 2.3: Beispiel für Rekombination in ES: Diskret auf den Objektvariablen, intermediär auf n Schrittweiten.

Bezeichnen wir wieder die aktuelle Generation mit $P^t := (p_1^t, \ldots, p_\mu^t)$, dann sind die Individuen p_i^t von der Form

$$p_i^t \;:=\; \Big((x_1^t, \ldots, x_n^t), (\sigma_1^t, \ldots, \sigma_s^t)\Big)\,, \tag{2.11}$$

wobei die Anzahl der Schrittweiten s meist 1 oder n ist.

Im folgenden wird der Übergang von einer Generation zur nächsten in einer ES beschrieben, bei der jeweils zwei Eltern einen Nachkommen erzeugen. Die Erweiterung auf mehrere Eltern bis hin zur *globalen Rekombination*, bei der alle Individuen der Elterngeneration bei der Erschaffung jedes Nachkommen beteiligt sind, ist dann offensichtlich.

Für $i = 1 \dots \lambda$:

Wähle p_{e1} und p_{e2} zufällig aus P^t

$p'^{\,t}_i = (\mathtt{rek}_x(p_{e1}, p_{e2}), \mathtt{rek}_\sigma(p_{e1}, p_{e2}))$

$p''^{\,t}_i = \mathtt{mut}(p'^{\,t}_i)$

Falls Plus-Strategie:

$P'''^{\,t} = \mathtt{sort}(P''^{\,t} \cup P^t)$

Sonst (Komma-Strategie):

$P'''^{\,t} = \mathtt{sort}(P''^{\,t})$

$P^{t+1} = (p'''^{t}_1, \dots, p'''^{t}_\mu)$.

Dabei wird häufig $\mathtt{rek}_\sigma$ intermediär und $\mathtt{rek}_x$ diskret gewählt. Der Operator `sort` erzeugt eine nach Fitneß sortierte Permutation seines Arguments. Der Mutationsoperator `mut` variiert die x'_i und σ'_i folgendermaßen:

$$\alpha \quad = \quad \tau_0 \cdot \mathrm{N}(0,1)\ . \tag{2.12}$$

Für $i = 1 \dots n$:

$$\sigma''^{\,t}_i \quad = \quad \sigma'^{\,t}_i \cdot e^{\alpha + \tau_1 \cdot \mathrm{N}(0,1)} \tag{2.13}$$

$$x''^{\,t}_i \quad = \quad x'^{\,t}_i + \sigma''^{\,t}_i \cdot \mathrm{N}(0,1)\ . \tag{2.14}$$

Da die Schrittweiten nicht negativ werden dürfen, wird ein multiplikativer Prozeß zur Variation verwendet. Man benötigt also eine Zufallsvariable mit einem Median von 1 und gleicher Wahrscheinlichkeitsdichte für y und $1/y$, $y \in \mathbb{R}^+$. Genau dies leistet $e^{\mathrm{N}(0,1)}$, die logarithmische Normalverteilung.

Zur Mutation der Schrittweiten wird je Individuum zunächst ein gemeinsamer Anteil α für alle Schrittweiten normalverteilt mit Standardabweichung τ_0 bestimmt. Zu diesem wird für jede einzelne Schrittweite noch eine weitere normalverteilte Zufallszahl mit Standardabweichung τ_1 ermittelt. Mit τ_0 und τ_1 kann man also die Variabilität der Schrittweitenanpassung einstellen und hat somit eine Entsprechung zur Mutationsrate der Genetischen Algorithmen, wobei ES deutlich weniger empfindlich auf eine schlechte Wahl dieser Parameter reagieren.

Korrelierte Mutationen

Dadurch, daß die Parameter jeweils mit Hilfe einer eindimensionalen Normalverteilung mit individueller Standardabweichung variiert werden, entsteht als Gesamtverteilung nicht etwa eine beliebige n-dimensionale Normalverteilung. Vielmehr liegen alle Punkte, die mit gleicher Wahrscheinlichkeit durch Mutation von einem gegebenen Punkt aus entstehen können, auf Hyperellipsoiden um diesen Punkt, die parallel zu den Koordinatenachsen ausgerichtet sind. Hierdurch werden achsenparallele Suchrichtungen, die in der Regel nicht optimal sind, bevorzugt.

Zur Vermeidung dieses Problems kann eine n-dimensionale Normalverteilung verwendet werden, so daß die Punkte, die mit gleicher Wahrscheinlichkeit Ergebnis einer Mutation sind, auf einem beliebig im Raum orientierten Hyperellipsoiden um den jeweiligen Ausgangspunkt liegen.

Will man eine n-dimensionale Normalverteilung verwenden, so benötigt man nicht nur n Standardabweichungen (bzw. die entsprechenden Varianzen), sondern auch $\frac{n}{2} \cdot (n-1)$ Kovarianzen. Anstatt nun die Kovarianzmatrix als weiteren Satz von Strategievariablen aufzunehmen, verwendet Schwefel die Lagewinkel eines n-dimensionalen Hyperellipsoiden. Während nämlich Veränderungen der Kovarianzen dazu führen können, daß die Matrix nicht mehr positiv definit ist, kann aus den Lagewinkeln immer eine positiv definite Kovarianzmatrix berechnet werden.

Wegen des hohen, quadratisch mit der Anzahl der Parameter steigenden Speicherplatzbedarfs und der zusätzlichen Berechnungen lohnt sich der Einsatz der korrelierten Mutationen vor allem dann, wenn die Zielfunktionsauswertung relativ lange dauert und damit der zusätzliche Aufwand nicht ins Gewicht fällt.

2.3.4 Evolutionäres Programmieren

Als dritter, ebenfalls zunächst unabhängig entwickelter Zweig der Evolutionären Algorithmen gilt das Evolutionäre Programmieren (EP), das auf Fogel, Owens und Walsh (1966) zurückgeht. Dort wurde EP als ein Verfahren zur Entwicklung endlicher Automaten mit vorgegebenem Ein-/Ausgabeverhalten definiert. Erst in den frühen 90er Jahren wurde EP zu einem Verfahren zur reellwertigen Parameteroptimierung weiterentwickelt (Fogel 1992). Da das biologische Modell von EP die Evolution von Arten ist, ist keine Rekombination vorgesehen. Die Mutation erfolgt bei reellwertiger Problemrepräsentation ähnlich wie bei der Evolutionsstrategie, einschließlich der Möglichkeit zur Selbstanpassung von Strategieparametern. Eine spezielle Form der Turnierselektion (s. Abschnitt 5.2.3) sorgt für den notwendigen Auswahlprozeß.

2.3.5 Varianten der Basisalgorithmen

In der Praxis finden sich heute überwiegend Varianten der Basisalgorithmen, die auf die jeweilige Anwendung zugeschnitten sind. Dies betrifft besonders die interne Repräsentation der möglichen Lösungen, da diese sich bei nichtakademischen Anwendungen, den sogenannten Real-World-Problemen, oft nur sehr umständlich auf Vektoren aus $\mathbb{R}^n$ oder $\mathbb{B}^l$ abbilden lassen. Stattdessen wird dann eine problemspezifische Repräsentation gewählt, zu welcher dann auch ein Mutations- und gegebenenfalls ein Rekombinationsoperator definiert werden muß. Beispiele für problemspezifische Operatoren finden sich bei Michalewicz (1996).

Beim GA wird vor allem die proportionale Selektion aufgrund ihrer oben genannten Nachteile nur noch selten verwendet. Stattdessen werden rangbasierte Selektionsverfahren eingesetzt, wie etwa die Linear-Ranking-Selektion (Baker 1985), die gewissermaßen eine Ordinalisierung der proportionalen Selektion darstellt, oder der Breeder-GA (Mühlenbein und Schlierkamp-Voosen 1993), der eine Variante der Komma-Selektion auf Genetische Algorithmen übeträgt.

Bei den ES wird in neueren Veröffentlichungen (Schwefel und Rudolph 1995) häufig auch noch die Alterung von Individuen zur Standard-ES hinzugezählt. Dies ermöglicht es, zwischen der potentiell unendlichen Lebensdauer von Individuen bei der $(\mu + \lambda)$-ES und der maximalen Lebensdauer von einer Generation bei der (μ, λ)-ES auch jede andere nichtnegative Anzahl von Generation als maximale Lebenszeit zuzulassen. Selektiert wird jeweils aus den Nachkommen und denjenigen Eltern, die das Höchstalter noch nicht überschritten haben.

2.3.6 ES+GA+EP+... = Evolutionäre Algorithmen

In Abschnitt 2.3.1 wurde bereits auf die algorithmische Ähnlichkeit zwischen den historischen EA hingewiesen. Schon früh wurde erkannt, daß die einzelnen Verfahren auch darüber hinaus sehr viele Gemeinsamkeiten aufweisen (Hoffmeister und Bäck 1991). Interne Repräsentationen können ebenso ausgetauscht werden wie die darauf definierten Operatoren.

In zunehmendem Maße gibt es Bestrebungen, diese unter dem gemeinsamen Namen *Evolutionäre Algorithmen* (Evolutionary Algorithms, EA) zusammenzufassen und das Forschungsgebiet als *Evolutionäres Rechnen* (Evolutionary Computation, EC) zu bezeichnen. Das Zusammenwachsen der Forschungsgemeinden führt auch dazu, daß theoretische Aussagen weitgehend vereinheitlicht werden können, wie z.B. bei Rudolph (1997) bezüglich der Konvergenzeigenschaften von EA oder bei Blickle (1998), wo Selektionsmechnismen unabhängig von ihren Ursprungsverfahren analysiert werden.

Kapitel 3

Räumliche Strukturen in parallelen EA

3.1 Historie der parallelen EA

Lange bevor Evolutionäre Algorithmen im heutigen Sinne bekannt waren, diente die inhärente Parallelität biologischer Systeme als Vorbild für effiziente Computermodelle. So entwickelte bereits von Neumann (1958) ein Konzept für einen zellular aufgebauten Parallelrechner. Jedoch war das einfachere, heute als von-Neumann-Rechner bekannte Rechnermodell mit nur einem Prozessor mit den damaligen technischen Möglichkeiten vergleichsweise einfach zu realisieren.

Die Entwicklung der Evolutionären Algorithmen fiel somit in eine Zeit, in der Parallelrechnerkonzepte zwar bekannt waren, aber eine hohe Verfügbarkeit massiv paralleler Computer nicht für absehbar gehalten wurde. Dies führte dazu, daß das inhärent parallele Populationsmodell zunächst sequentialisiert wurde.

Mit der ansteigenden Präsenz von Parallelrechnern vor allem in den 80er Jahren entstanden zwei Arten von parallelen EA. Auf der einen Seite wurden die klassischen sequentiellen Verfahren parallelisiert, auf der anderen Seite entstanden parallele Varianten auf der Basis biologischer Paradigmen. Zu letzteren wurden dann häufig Speed-Up-Betrachtungen angestellt. Da aber Parallelisierung im strengen Sinne bedeutet, daß man zu einem sequentiellen Algorithmus eine parallele Variante mit identischem Ein-Ausgabeverhalten entwickelt, ist in diesem Falle ein

Laufzeitvergleich nicht zulässig. Insbesondere kann man jeden parallelen Algorithmus wieder sequentialieren, so daß ein superlinearer Speed-Up unmöglich ist.

In der Biologie nennt man den Zustand einer Population, bei dem jedes Individuum potentiell mit jedem anderen (passenden Geschlechts) Nachkommen haben kann, *Panmixie* (Sedlag und Weinert 1987). Da die Parallelisierung Evolutionärer Algorithmen in den meisten Fällen zu einer Aufteilung der Population (s. Abschnitt 3.3) führte, wurde auch von nichtpanmiktischen Populationen gesprochen.

3.2 Klassifizierung nach Gorges-Schleuter

3.2.1 Ursprüngliche Klassifizierung

Die erste Klassifizierung paralleler Evolutionärer Algorithmen stammt von Martina Gorges-Schleuter (Gorges-Schleuter 1990). Sie definierte die folgenden drei Klassen:

- Island Model:
  Die Population setzt sich aus voneinander getrennten, in sich panmiktischen Teilpopulationen zusammen. Genetisches Material wird in begrenztem Maße zwischen beliebigen Teilpopulationen ausgetauscht.
- Stepping Stone Model:
  Die Population setzt sich aus voneinander getrennten, in sich panmiktischen Teilpopulationen zusammen. Auf den Teilpopulationen ist eine Nachbarschaftsrelation definiert. Genetisches Material wird in begrenztem Maße nur zwischen benachbarten Teilpopulationen ausgetauscht.
- Neighborhood Model:
  Es existiert eine einzige Population. Auf den Individuen ist eine Nachbarschaftsrelation definiert. Individuen interagieren nur mit ihren Nachbarn.

3.2.2 Verfeinerung der Klassifizierung

Man sieht sofort, daß die Trennung zwischen Island Model und Stepping Stone Model fallengelassen werden kann, wenn man als Nachbarschaftsrelation auch die totale Relation zuläßt. Häufig wird dann von einem *Migrationsmodell* (Rudolph 1990) gesprochen. Allerdings impliziert dieser Begriff auch bereits die Art des Austauschs von Informationen: In Anlehnung an Migration in der Natur verlassen die Migranten ihre bisherige Population und wechseln in eine benachbarte. Viel häufiger werden aber in Evolutionären Algorithmen nur Kopien der ausgewählten Migranten versandt. Dies soll im folgenden, dem Bild ortsfester Pflanzen entsprechend, die Kopien ihrer Gene in Form von Pollen aussenden, *Pollinationsmodell* genannt werden. Ein ähnlicher Begriff (plant pollination model), den Goldberg (1989) für einen feinkörnigen Ansatz vorgeschlagen hat, hat keine Verbreitung gefunden.

Die Mehrzahl der Nachbarschaftsmodelle ist in einem Euklidischen Raum definiert und greift auf dessen Abstandsdefinition zurück. Wir wollen diese *metrische Nachbarschaftsmodelle* nennen. Eine aktuellere Fassung der obigen Klassifizierung könnte wie folgt aussehen:

- Panmiktisches Modell (panmictic model):
  Alle Individuen stehen in direkter Konkurrenz, ihre Nachkommen in der nächsten Generation zu plazieren.
- Multipopulationsmodelle (multipopulation models):
  - Migrationsmodell (migration model):
    Die Population setzt sich aus voneinander getrennten, in sich panmiktischen Teilpopulationen zusammen. Genetisches Material wird in begrenztem Maße zwischen Teilpopulationen ausgetauscht, die als benachbart definiert sind. Beim Austausch werden Informationen verschoben, nicht kopiert.
  - Pollinationsmodell (pollination model):
    Die Population setzt sich aus voneinander getrennten, in sich panmiktischen Teilpopulationen zusammen. Genetisches Material wird in begrenztem Maße zwischen Teilpopulationen ausgetauscht, die als benachbart definiert sind. Beim Austausch werden Informationen kopiert.
- Nachbarschaftsmodelle (neighborhood models):
  - Metrisches Nachbarschaftsmodell (metric neighborhood model):
    Es existiert eine einzige Population von räumlich angeordneten Indi-

viduen. Über den Abstand von Individuen ist eine Nachbarschaftsrelation definiert. Individuen interagieren nur mit ihren Nachbarn.

– Relationales Nachbarschaftsmodell (relational neighborhood model): Es existiert eine einzige Population. Auf den Individuen ist eine Nachbarschaftsrelation definiert. Individuen interagieren nur mit ihren Nachbarn.

Auch diese Klassifizierung ist weder vollständig noch überschneidungsfrei. Sie dient lediglich der groben Einordnung von Populationsstrukturen. Insbesondere läßt sich mit dem relationalen Nachbarschaftsmodell jedes der anderen in einfacher Weise beschreiben — was wir dann im Kapitel 4 auch tun werden.

3.3 Räumliche strukturierte Populationsmodelle

Im folgenden wird versucht, eine Übersicht über bisher veröffentlichte Ansätze zur Parallelisierung Evolutionärer Algorithmen mit strukturierten Populationen zu geben. Die meisten der hier aufgeführten Arbeiten stammen aus den späten 80er und frühen 90er Jahren. Dies liegt daran, daß in dieser Zeit alle wesentlichen Techniken entwickelt wurden, während nachfolgend entweder die Anwendung oder die Analyse der bestehenden Verfahren im Mittelpunkt stand.

3.3.1 Panmiktische Teilpopulationen

Ein früher Ansatz eines Multipopulationsmodells stammt von Cohoon, Hedge, Martin und Richards (1987). Die biologische Motivation dieses Ansatzes beruht auf der Theorie der *Punctuated Equilibria* (PE), deren wesentliche Argumente auf der Annahme beruhen, daß Artenbildung überwiegend durch Veränderung der Umwelt hervorgerufen wird, und, umgekehrt, bei konstanter Umwelt ein Gleichgewichtszustand erreicht wird, in dem sich eine Art nur langsam durch genetische Drift verändert. Die Autoren haben diese Theorie auf EA zu übertragen versucht, indem sie räumlich getrennte Teilpopulationen nach mehrere Generation dauernden Isolationsphasen (*Epochen*), die der Erreichung des Gleichgewichtszustandes dienen sollen, mit Nachbarpopulationen Individuen austauschen lassen. Letzteres soll eine veränderte Umwelt gemäß der PE-Theorie modellieren und den Speziationsprozeß wieder in Gang setzen.

Zur gleichen Zeit präsentierten Pettey, Leuze und Grefenstette (1987) ebenfalls ein Multipopulationsmodell unter dem Namen *Parallel Genetic Algorithm* (PGA). Als biologische Motivation wird der Begriff der *polytopischen Spezies* genannt, also einer Art mit weitgehend isolierten Teilpopulationen, die aber weiterhin untereinander rekombinationsfähig sind. Im Unterschied zum Punctuated Euquilibria spielen Isolationszeiten bei diesem Ansatz keine Rolle. Ein Versuch einer theoretischen Analyse findet sich bei Pettey und Leuze (1989). Betrachtet wird die Änderung der Wirkung der Operatoren gegenüber einer panmiktischen Variante.

Ein Ansatz von Tanese (1987) ist direkt auf die verfügbare Hardware zugeschnitten, einen als vierdimensionalen Hyperwürfel verschalteten Mehrprozessorrechner. Dementsprechend kommunizieren jeweils Teilpopulationen miteinander, die durch eine Kante des Hyperwürfels verbunden sind. Isolationszeiten werden an einem Beispiel untersucht.

Starkweather, Whitley und Mathias (1991) parallelisierten das GENITOR-Konzept (Whitley 1989), bei welchem in jeder Generation nur eine Individuum ersetzt wird.

3.3.2 Nachbarschaftsmodelle

Das von Gorges-Schleuter (1989) entwickelte ASPARAGOS-Modell ist ein metrisches Nachbarschaftsmodell auf einem Torus der Höhe 2. Dadurch entsteht eine Leiterstruktur, bei der jedes Individuum drei direkte Nachbarn hat. Dies war nicht zuletzt durch die verfügbare Hardware motiviert, denn die Transputer von Inmos, die als Rechenknoten verwendet wurden, verfügten über vier bidirektionale, serielle Links, von denen bei ASPARAGOS einer für die Kommunikation mit einer Datenhaltungskomponente übrigblieb. Mit ASPARAGOS wurden sehr gute Ergebnisse auf Traveling-Salesperson-Problemen erzielt. Durch die Verwendung von gerichteter Mutation mittels deterministischer Greedy-Optimierungsschritte kann der Einfluß der Parallelisierung nicht einzeln bewertet werden.

Parallele Genetische Algorithmen auf zweidimensionalen Gitterstrukturen wurden auch von Spiessens und Manderick (1989) und Davidor (1991) vorgeschlagen. Letzterer verwendet keine Partnerselektion, sondern eine probabilistische Ersetzungsstrategie der Eltern durch ihre Nachkommen, die an nichtdeterministische zelluläre Automaten erinnert.

3.3.3 Gemischte parallele Populationsmodelle

Kapsalis, Rayward-Smith und Smith (1994) stellen ein *vereinheitlichtes Paradigma* für parallele GA vor. Ausgehend von Goldbergs Simple GA (SGA) entwickelten sie eine GA-Variante mit parametrisierbarer Populationsstruktur. Dabei wird zwischen zwei Ebenen der Parallelität unterschieden: Parallele Zielfunktionsauswertung (Level-1) und parallele Reproduktion (einschließlich Zielfunktionsauswertung, Level-2). Das vereinheitlichte Paradigma erschöpft sich schließlich in einer Implementierung, die sich zwischen den Varianten Master-Slave (Level-1), Island-Model und Nachbarschaftsmodell (Level-2) umschalten läßt. Allerdings wird auch in den Conclusions die bessere Vergleichbarkeit der verschiedenen Ansätze als wesentliches Ergebnis dieser Arbeit dargestellt.

Bei Diogenes (Sprave 1990; Hoffmeister, Rudolph und Sprave 1991) wird eine auf einem Torus angeordnete Population parallelisiert, indem rechteckige Teilgitter auf Prozessoren verteilt werden. Zwischen den angrenzenden Teilpopulationen werden Kopien der Randindividuen ausgetauscht, um die Nachbarschaften der lokalen Populationen zu vervollständigen. Über die Größe der lokalen Populationen kann man die Rechenlast auf den einzelnen Prozessoren anpassen. Durch Isolationsphasen, während derer die Kopien der Randindividuen nicht aktualisiert werden, ist auch die Kommunikationslast parametrisierbar, so daß ein Nachbarschaftsmodell mit skalierbarer Parallelität entsteht.

3.4 Analytische Untersuchungen von Populationsstrukturen

Es gibt bereits mehrere Ansätze für eine theoretische Analyse nichtpanmiktischer Evolutionärer Algorithmen. In den meisten Fällen werden aber nur sehr spezielle Topologien untersucht. Nur Gordon (1994) liefert einen Ansatz, der nicht nur grob- und feinkörnig parallele Algorithmen berücksichtigt, sondern auch den panmiktischen Fall mit Turnierselektion. Ziel dieser Arbeit war eine Definition von Lokalität in EA, die auf möglichst viele räumlich strukturierte Populationsmodelle anwendbar ist. Gordon verwendet hierzu den prozentualen Anteil von nichtlokalen Zugriffen auf Individuen im Verhältnis zur Gesamtanzahl der Zugriffe. Dieser Wert muß für jeden Selektionsoperator und jede Topologie einzeln bestimmt werden.

Cantú-Paz und Goldberg (1997) vergleichen als Grenzfälle des Inselmodells den Fall maximaler Migrationsfrequenz und Migrantenzahl mit dem Fall nicht interagierender Teilpopulationen und kommt zu dem Ergebnis, daß mit Migration dieselbe Ergebnisqualität mit einer geringeren Anzahl an Auswertungen erreicht werden kann. In kürzlich erschienenen internen Berichten untersucht Cantú-Paz vor allem die Wirkung von Kommunikationstopologien und Austauschmechanismen beim Migrationsmodell.

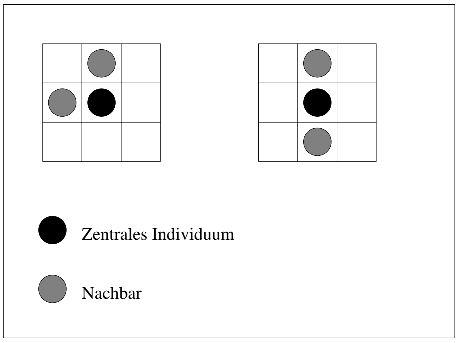

Abbildung 3.1: Skizze zweier Nachbarschaften mit gleichem Radius nach Sarma (1998). Die linke Nachbarschaft spannt den Raum auf, so daß ein zusammenhängender Kommunikationsgraph entsteht. Die rechte Nachbarschaft erzeugt für jede Spalte des Torus einen isolierten Teilgraphen.

In ihrer Dissertation untersucht Sarma (1998) ein Maß für den Selektionsdruck auf torusbasierten, metrischen Nachbarschaftsmodellen. Dazu wird der *Nachbarschaftsradius*[1] definiert als

$$rad((x,y),N(x,y)) = \sqrt{\frac{\sum_{i=1}^{n}(x_i-x)^2+\sum_{i=1}^{n}(y_i-y)^2}{n}} \quad (3.1)$$

wobei (x,y) die Gitterkoordinaten des Zentrums einer Nachbarschaft $N(x,y) = \{(x_i,y_i)|i=1,\ldots,n\}$ sind. Die Problematik dieses geometrisch inspirierten Ansatzes läßt sich an einem einfachen Beispiel aufzeigen: Die in Abb. 3.1 skizzierten Nachbarschaften weisen denselben Radius auf, obwohl die linke Nachbarschaft

[1] Diese Definition des Nachbarschaftsradius' weicht von der in Kap. 4 verwendeten Definition nach Sprave (1994) ab.

den ganzen Torus aufspannt, während die rechte Nachbarschaft die Population in voneinander getrennte, ringförmige Teilpopulationen zerfallen läßt.

Anhand von Evolutionsstrategien mit ring- und torusbasierten Nachbarschaftsmodellen untersucht Gorges-Schleuter (1998) die Wirkung von *Selfing*, also der Ausschaltung von Rekombination durch Mehrfachselektion eines einzigen Elters für einen Nachkommen. Es wird gezeigt, daß dieser Fall bei feinkörnigen Modellen, bedingt durch die typischerweise kleine Anzahl von Nachbarn, sehr häufig auftritt. Zusätzlich werden experimentell Hinweise darauf geliefert, daß Selfing vorzeitige Stagnation in Nachbarschaftsmodellen begünstigt.

Kapitel 4

Vereinheitlichte Modellierung von Populationsstrukturen

4.1 Motivation

Im folgenden Abschnitt wird eine Modellierung von Populationsstrukturen vorgestellt, die fein- und grobkörnig strukturierte EA in einer einheitlichen Notation erfaßt. Diese Modellierung soll vor allem als Grundlage für theoretische Betrachtungen dienen. Auf der anderen Seite erlaubt sie aber auch die genaue Spezifizierung von angewandten Populationsmodellen und erleichtert somit die Dokumentation nachvollziehbarer Experimente.

4.2 Hypergraphen

Während man in der klassischen Graphentheorie Kanten in ungerichteten Graphen als maximal zweielementige Teilmengen der Knotenmenge definiert, läßt man bei Hypergraphen Teilmengen beliebiger, von Null verschiedener Größe zu. Die Theorie der Hypergraphen geht zurück auf Claude Berge (1973).

Hypergraphen stellen somit eine Erweiterung des klassischen Graphenbegriffs dar, die sich immer dann als Modellierungsansatz anbieten, wenn gleichartige Beziehungen zwischen mehreren Dingen formalisiert werden sollen, etwa im Schaltungsentwurf (Raith 1994), in der Simulation zellularer Strukturen (Hartmann 1994) oder zur Definition von Software-Metriken (Stetter 1989).

Ein interessanter Modellierungsansatz der biologischen Mutation und Rekombination mittels Hypergraphen findet sich bei Gitchoff und Wagner (1996).

Def. 1 Sei $X = \{x_1, x_2, \ldots, x_n\}$ eine endliche Menge, und sei $\mathcal{E} = (E_i \mid i \in I)$ eine Familie von Teilmengen von X mit der Indexmenge I. Die Familie $\mathcal{E}$ wird *Hypergraph auf* X genannt, wenn

$$E_i \neq \emptyset \; \forall i \in I \tag{4.1}$$

$$\bigcup_{i \in I} E_i = X \; . \tag{4.2}$$

Entsprechend wird das Paar $H = (X, \mathcal{E})$ Hypergraph genannt, und $|X|$ die *Ordnung* des Hypergraphs. Die Elemente von $\mathcal{E}$ heißen *Kanten* des Hypergraphs. Ein Hypergraph, in welchem alle Kanten paarweise verschieden sind, wird *einfacher* Hypergraph genannt.

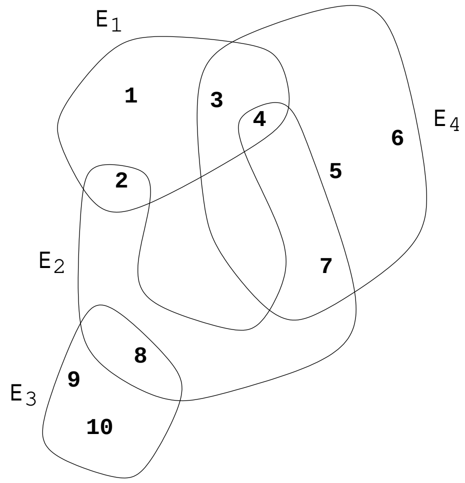

Abbildung 4.1: Ein Hypergraph mit Knoten $X = \{1, \ldots, 10\}$ und Kanten $E_1 = \{1, 2, 3, 4\}$, $E_2 = \{2, 4, 7, 8\}$, $E_3 = \{8, 9, 10\}$ und $E_4 = \{3, 4, 5, 6, 7\}$.

Zwei Knoten x_k und x_l eines Hypergraphs heißen *adjazent*, wenn sie einer gemeinsamen Kante angehören, also

$$\exists i \in I : \quad x_k \in E_i \wedge x_l \in E_i \; . \tag{4.3}$$

Zu jedem Hypergraph $H = (X, \mathcal{E})$ existiert ein *dualer Hypergraph* $H^* = (E, \mathcal{X})$, mit Knoten $e_1, \ldots, e_m$ und Kanten $\{X_1, \ldots, X_n\}$, wobei gilt:

$$\forall j \in \{1, \ldots, n\} : \quad X_j = \{e_i \mid i \leq m,\ E_i \ni x_j\} \; . \tag{4.4}$$

Dabei korrespondieren die Knoten e_i mit den Kanten E_i von H und die Kanten X_j mit den Knoten x_j von H. Anschaulich formuliert werden also zu jedem Knoten aus H alle Kanten, die ihn enthalten, zu einer Kante des dualen Hypergraphs zusammengefaßt.

4.2.1 k-Schnitte von Hypergraphen

k-Schnitte von Hypergraphen entstehen durch Reduzierung des Rangs auf eine positive Zahl k, indem alle Hyperkanten der Mächtigkeit $|E_i| > k$ durch alle Teilmengen der Mächtigkeiten k ersetzt werden, formal: Gegeben sei ein Hypergraph $H = (X, \mathcal{E})$ und eine natürliche Zahl $k > 0$. Dann ist der k-Schnitt von H der Hypergraph $H_{(k)} = (X, \mathcal{E}_{(k)})$ mit

$$\begin{aligned} \mathcal{E}_{(k)} \;&=\; \left\{F \subseteq X \;\middle|\; |F| = k \wedge \exists E_i \in \mathcal{E} :\ F \subseteq E_i\right\} \\ &\quad\; \cup \left\{F \subseteq X \mid\ |F| < k \wedge \exists E_i \in \mathcal{E} :\ F = E_i\right\} \; . \end{aligned} \tag{4.5}$$

Von besonderem Interesse ist dabei der 2-Schnitt $H_{(2)}$ eines Hypergraphs H. Gemäß Definition ist $H_{(2)}$ ein Hypergraph vom Rang 2, in welchem jeder Knoten mit einer Schlinge versehen ist. Mit $(H)_2$ sei dann $H_{(2)}$ ohne Schlingen bezeichnet. Offensichtlich ist $(H)_2$ ein einfacher Graph, in welchem 2 Knoten adjazent sind, wenn es mindestens eine Kante von H gibt, in der beide enthalten sind. Folglich bilden die Kanten aus H Cliquen in $(H)_2$.

Ein Hypergraph H besitzt die *Helly-Eigenschaft*, wenn der Durchschnitt über jede Teilmenge der Kantenmenge $\mathcal{E}$ mit paarweise nichtleerem Durchschnitt ebenfalls nicht leer ist:

$$\mathcal{E}' \subseteq \mathcal{E} \ \wedge \ \forall E'_i, E'_j \in \mathcal{E}' : \ E'_i \cap E'_j \neq \emptyset \ \Rightarrow \ \bigcap_{E' \in \mathcal{E}'} \neq \emptyset \tag{4.6}$$

Zu jedem Hypergraph $H = (X, \mathcal{E})$ wird der einfache Graph $L(H) = (V, E), |V| = |X|$ mit

$$(e_i, e_j) \in V \quad \Leftrightarrow \quad E_i \cap E_j \neq \emptyset, \ i, j = 1, \ldots, |X| \tag{4.7}$$

der *repräsentative Hypergraph*[1] von H genannt. Die Knoten des repräsentativen Graphs entsprechen den Kanten von H. Der repräsentative Graph beschreibt also die Inzidenzrelation auf den Kanten des Hypergraphs.

4.2.2 Repräsentationen von Hypergraphen

Die Effizienz von Algorithmen auf einfachen Graphen beruht meist auf einer geeigneten Repräsentation eines gegebenen Graphs. Häufig werden Algorithmen auf Adjazenzlisten[2] oder Adjazenzmatrizen definiert. Die Übertragung dieser Darstellungen auf Hypergraphen ist nicht problemlos, da die Information, über welche Kante zwei Knoten adjazent sind, verloren geht.

Eine verlustfreie Darstellung eines Hypergraphs ist seine Inzidenzmatrix. Dabei repräsentieren die Spalten der Matrix die Kanten des Hypergraphs und die Zeilen die Knoten. Sei H ein Hypergraph mit m Kanten und n Knoten, dann wird er durch die Matrix A mit

$$a_{i,j} = \begin{cases} 1 & \text{falls} \quad x_i \in E_j \\ 0 & \text{sonst} \end{cases} \tag{4.8}$$

[1] Engl. representative graph. Im dreisprachigen Index in (Berge 1973) wird als deutsche Übersetzung *entsprechender Graph* angegeben.

[2] Falls nicht anders angegeben, ist mit Adjazenz die Adjazenz von Knoten gemeint.

eindeutig beschrieben. Die Inzidenzmatrix des Hypergraphs in Abb. 4.1 ist

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{4.9}$$

Die Adjazenz zweier Knoten x_i und x_k kann dann mit einer Zeitkomplexität von $O(|\mathcal{E}|)$ als Skalarprodukt der i-ten und k-ten Zeile berechnet werden:

$$< a_{i,\cdot}, a_{k,\cdot} >= 1 \quad \Leftrightarrow \quad \exists E_j : x_i \in E_j \ \wedge \ x_k \in E_j . \tag{4.10}$$

Die Adjazenzmatrix B mit $b_{i,k} =< a_{i,\cdot}, a_{k,\cdot} >$ erhält man dann in $O(|\mathcal{E}| \cdot |X|)$:

$$B = AA^\top . \tag{4.11}$$

Die Matrix B ist gerade die Adjazenzmatrix des 2-Schnitts $H_{(2)}$ von H. Entsprechend kann man die Adjazenz von Kanten als Skalarprodukt der zugehörigen Zeilen der Inzidenzmatrix ausdrücken. Man erhält als Kantenadjazenzmatrix die Matrix

$$B^* = A^\top A , \tag{4.12}$$

also ebenfalls in $O(|\mathcal{E}| \cdot |X|)$. Die Matrix B^* ist die Knotenadjazenzmatrix des dualen Hypergraphs H^* und damit auch die Adjazenzmatrix des repräsentativen Graphs $L(H)$.

Es stellt sich nun die Frage, ob die oben definierten Adjazenzmatrizen eine geeignete Basis für Hypergraphenalgorithmen darstellen. So erlaubt zum Beispiel die Kantenadjazenzmatrix B^* die Berechnung des Durchmessers Δ_H eines Hypergraphs $H = (X, \mathcal{E}), |X| = n$ durch

$$\Delta_H = min \left\{ k \;\middle|\; \sum_{i=0}^{k} B^{*i} = K_n \right\} \tag{4.13}$$

wobei mit K_n die Adjazenzmatrix des vollständigen Graphs mit n Knoten bezeichnet ist. Bei der Betrachtung von Populationsstrukturen sind vor allem die Matrizen mit *verallgemeinerten Koeffizienten* hilfreich. Diese entstehen dadurch, daß die Matrizenmultiplikationen nicht in $\mathbb{B}$, sondern in $\mathbb{N}$ oder $\mathbb{Z}$ durchgeführt werden. Sei etwa A die Adjazenzmatrix eines einfachen Graphs $G = (V, E)$. Dann geben beispielsweise die verallgemeinerten Koeffizienten $a^k_{i,j} \in \mathbb{N}$ der Matrix A^k die Anzahl der Pfade der Länge k zwischen v_i und v_j, und die Elemente der Matrix

$$\hat{A} = \sum_{k=1}^{|V|} A^k \tag{4.14}$$

die Anzahl aller Pfade beliebiger Länge zwischen den jeweiligen Knoten an.

4.2.3 Modellierung von Populationsstrukturen mittels Nachbarschaftsgraphen

Nichtpanmiktische Populationsmodelle werden in der Literatur meistens auf regelmäßige geometrische Strukturen zurückgeführt. Besonders häufig trifft man Ringe, Tori und Würfel bzw. Hyperwürfel an, welche allesamt auf Gitterstrukturen definiert werden können. All diesen Ansätzen ist gemeinsam, daß man sofort ein anschauliches Abstandsmaß zur Verfügung hat. Der zweidimensionale Torus, die Kombination aus einfacher Anschauung — zweidimensionales Gitter — und einfacher Implementierung — Eliminierung von Gittergrenzen durch Modulo-Rechnung — ist dabei die beliebteste Anordnung von Individuen bei feinkörnigen Modellen.

Eine wie auch immer verwendete Nachbarschaft zwischen Individuen wird dabei in der Regel nicht auf benachbarte Gitterpunkte beschränkt, sondern über ein

Abstandsmaß auf dem Gitter definiert. Hierzu werden wiederum klassische Gittermetriken herangezogen: Die sogenannte Manhattan-Metrik basierend auf der L_1-Norm und die Anzahl der besuchten Gitterpunkte, die auf die L_∞-Norm zurückgeführt werden kann.

Letztere benötigt noch nicht einmal die Einbettung in einen Euklidischen Vektorraum, sondern kann einfach als Abstand in einem Graph betrachtet werden. An dieser Stelle werden wir aber auf eine formale Definition dieser grapheninduzierten Populationsstrukturen verzichten, da sie sich in die nachfolgende, allgemeinere Definition auf elegante Weise abbilden lassen.

4.2.4 Modellierung von Populationsstrukturen mittels Hypergraphen

In den Kapiteln 3.3.1 und 3.3.2 sind eine Reihe von Ansätzen zu nicht-panmiktischen EA dargestellt. Um diese in einen gemeinsamen Kontext zu setzen, wird im folgenden ein Formalismus vorgestellt, der im Hinblick auf Informationsausbreitung einen direkten Vergleich verschiedener Ansätze für räumlich strukturierte EA ermöglicht.

Def. 2 Sei A der Suchraum eines gegebenen Optimierproblems, S der Raum der vom betrachteten Verfahren verwendeten Zustandsinformationen. Dann heißt ein Element der Menge $A \times S$ *Individuum* und eine Familie von Individuen $(p_0^{(t)}, \dots, p_{\lambda-1}^{(t)}),\ p_i^{(t)} \in A \times S$ *Population* zum Zeitpunkt t.

Zustandsinformationen sind z.B. die Schrittweiten und Lagewinkel der klassischen Evolutionsstrategie. Eine *Population zum Zeitpunkt* t wird häufig auch *Population in der* t*-ten Generation* oder einfach t-te Generation genannt.

Die tatsächlichen Werte der Individuen werden uns zunächst nicht interessieren. Für die Definition von Populationsstrukturen reicht es aus, Individuen mit ihren Indizes zu identifizieren. Daher definieren wir:

Def. 3 Eine *Populationsstruktur* Π auf einer Population P mit $|P| = \lambda$ ist ein Paar (H, Q), bestehend aus einem Hypergraph $H = (X, \mathcal{E})$, $X = \{\overline{0}, \dots, \overline{\lambda - 1}\}$, $\mathcal{E} \subseteq \mathcal{P}(X)$, und einer Partition $Q \subset \mathcal{P}(\mathcal{X})$ von X mit $|Q| = |\mathcal{E}|$. Die Hyperkante E_i heißt *Dem* (engl. *deme*) aller Elemente aus $Q_i \in Q$.

Da wir ausschließlich auf endlichen Populationen arbeiten werden, bietet es sich an, als Indexmenge den Restklassenkörper $\lambda SetZ$ zu verwenden. Aus Gründen der Lesbarkeit wird im weiteren Text jedoch auf die explizite Kennzeichnung der Restklassen verzichtet und $0, \ldots, \lambda - 1$ anstelle von $\overline{0}, \ldots, \overline{\lambda - 1}$ verwendet. Man beachte jedoch, daß, falls nicht anders gekennzeichnet, alle Berechnungen auf Indizes in Populationen modulo λ durchgeführt werden.

Diese gewissermaßen syntaktische Definition des Begriffes der Populationsstruktur ist natürlich mit der Intention einer bestimmten Semantik gewählt worden. Wie der Begriff *Dem* bereits nahelegt, soll die einer Teilmenge Q_i von Individuen zugeordnete Hyperkante E_i die Menge seiner potentiellen Eltern beim Übergang von der t-ten zur $t + 1$-ten Generation bilden.

Def. 4 Eine Populationsstruktur Π heißt

- *panmiktisch*, wenn $\mathcal{E} = \{X\}$,
- *reflexiv*, wenn $\forall x \in X, i = 0, \ldots, |\mathcal{E}| - 1 : x \in Q_i \Rightarrow x \in E_i$,
- *symmetrisch*, wenn gilt: $|E_i \cap Q_j| = |E_j \cap Q_i|$, und
- *reduzierbar* auf eine Populationsstruktur $\Pi' = (X, \mathcal{E}', \mathcal{Q}')$, wenn Π durch Weglassen mehrfacher Kanten aus $\mathcal{E}$ in Π' überführt werden kann.

Viele der gängigen nichtpanmiktischen EA lassen sich als reflexive und symmetrische Populationsstrukturen modellieren, so daß jedes Individuum potentieller Elter seines Nachfolgers an derselben Position in der Population ist (Reflexivität), und daß zwei benachbarte Teilpopulationen in beide Richtungen dieselbe Anzahl an Individuen austauschen (Symmetrie).

Def. 5 Zwei Populationsstrukturen heißen *äquivalent*, wenn sie bis auf Indexvertauschungen auf dieselbe Populationsstruktur reduzierbar sind.

Die Partition Q von X kann auch als Hypergraph mit disjunkten Kanten angesehen werden, und somit läßt sich auch die Inzidenzmatrix aufstellen. Damit wird es möglich, mit Hilfe einfacher Matrixoperationen Berechnungen auf Populationsstrukturen vorzunehmen, wie in Satz 1 gezeigt wird. In Kapitel 6 wird das

Takeover-Verhalten von Populationsstrukturen untersucht werden. Die Idee solcher Betrachtungen ist es, festzustellen, wieviele Generationen ein Superindividuum zur Übernahme der ganzen Population benötigt, wenn nur Selektion angewandt wird. Eine untere Grenze für die Takeover-Zeit ist offensichtlich die Anzahl von Generationen, die ein Superindividuum benötigt, um bei deterministischer Übernahme des Besten (Elitist-Selektion) die gesamte Population zu besetzen. Diese Elitist-Takeover-Zeit läßt sich aus der Definition der Populationsstruktur ableiten:

Satz 1 Gegeben sei eine Populationsstruktur $\Pi = (X, \mathcal{E}, \mathcal{Q})$. $\overline{E}$ und $\overline{Q}$ seien die Inzidenzmatrizen von $(X, \mathcal{E})$ und $(X, \mathcal{Q})$. Sei G der lineare Graph mit der Knotenadjazenzmatrix $\overline{E}\,\overline{Q}^{\top}$. Dann ist die Elitist-Takeover-Time von Π gleich dem Durchmesser von G.

Beweis: Seien $x, y \in X$, $N = |\mathcal{E}|$. Das Individuum x ist genau dann potentieller Elter von y, wenn gilt: $\exists \nu \in \{0, \dots, N-1\} : x \in E_\nu \cap y \in Q_\nu$. Also gilt $\sum_{i=0}^{N-1} e_{x,i} \cdot q_{i,y} =< e_{x,\cdot}, q_{\cdot,y} >= g_{x,y} = 1$. Sei nun $\overline{x} \in X$ ein bestes Individuum in X in der Generation t. Unter der Voraussetzung der Elitist-Selektion sind dann alle y mit $(\overline{x}, y) \in G$ ebenfalls beste Individuen in Generation $t+1$. Die maximale Anzahl an Generationen bis zu welcher alle $y \in X$ beste Individuen sind, ist daher gleich der Länge des längsten kürzesten Pfades in G $\Box$.

Def. 6 Der *Durchmesser* Δ_Π einer Populationsstruktur $\Pi = (X, \mathcal{E}, \mathcal{Q}), |X| = \lambda$ ist definiert als

$$\Delta_\Pi \quad := \quad min\left\{ k \in \mathbb{N} \cup \{\infty\} \mid \sum_{i=1}^{k} (\overline{E}\,\overline{Q}^{\top})^i = 1^{\lambda \times \lambda} \right\}, \tag{4.15}$$

Eine Populationsstruktur mit $\Delta_\Pi \neq \infty$ heißt *zusammenhängend*.

Nicht zusammenhängende Populationsstrukturen sind tatsächlich untersucht worden: Von Cantú-Paz und Goldberg (1997) werden sie als Grenzfall für Migrationsmodelle mit Isolation untersucht.

4.3 Hypergraphmodellierung klassischer Populationsmodelle

4.3.1 Panmixie

Das ursprüngliche Populationsmodell der EA ist panmiktisch:

$$\begin{aligned} \Pi_{\text{pan}} &= (X, \mathcal{E}, \mathcal{Q}) \\ X &= \{0, \ldots, \lambda - 1\} \\ \mathcal{Q} &= (X) \\ \mathcal{E} &= (X) \end{aligned} \tag{4.16}$$

4.3.2 Multipopulationsmodelle

Im folgenden wird nun die Modellierung eines Migrationsmodells mit r Teilpopulation von jeweils λ/r Individuen und zufälliger Auswahl von m Migranten und ohne Isolationszeiten vorgestellt.

Gegeben sei eine Population $P = \{0, \ldots, \lambda - 1\}$, $\lambda = r\nu,\ r, \nu \in \mathbb{N}$. Die Teilpopulationen des Migrationsmodells seien $Q_i = \{i\nu, \ldots, i\nu + \nu - 1\}, i = 0, \ldots, r-1$. Weiterhin sei $M_{s \to t} \subset Q_s, s, t \in r\mathbb{Z}$ die Menge der Migranten von Q_s nach Q_t. Ein Migrationspfad von Q_s nach Q_t existiert dann also für $M_{s \to t} \neq \emptyset$. Die Populationstruktur ergibt sich nun aus

$$\begin{aligned} \Pi_{\text{migr}} &= (X, \mathcal{E}, \mathcal{Q}) \\ X &= \{0, \ldots, \lambda - 1\} \\ Q_i &= \{i\nu, \ldots, i\nu + \nu - 1\}, \quad i = 0, \ldots, r-1 \\ \mathcal{E} &= (E_0, \ldots, E_{r-1}) \\ E_i &= Q_i \cup \textstyle\bigcup_{s=0}^{r-1} M_{s \to i} \setminus \bigcup_{t=0}^{r-1} M_{i \to t}, \\ & \qquad M_{s \to i} \subset Q_s,\ M_{i \to t} \subset Q_i \end{aligned} \tag{4.17}$$

Diese Populationsstruktur ist nicht reflexiv. Die Immigranten sind so modelliert, daß sie nicht Eltern der Individuen sein können, die in der folgenden Generation ihre Positionen einnehmen. Sind die $M_{s \to t}$ zusätzlich noch paarweise disjunkt, so ist $\mathcal{E}$ eine Partition von X. Läßt man zu, daß Individuen potentielle Eltern in mehreren Teilpopulationen sind, werden Informationen kopiert und man erhält ein Pollinationsmodell:

$$
\begin{aligned}
\Pi_{\text{poll}} &= (X, \mathcal{E}, \mathcal{Q}) \\
X &= \{0, \ldots, \lambda - 1\} \\
Q_i &= \{i\nu, \ldots, i\nu + \nu - 1\}, \quad i = 0, \ldots, r-1 \\
\mathcal{E} &= (E_0, \ldots, E_{r-1}) \\
E_i &= Q_i \cup \textstyle\bigcup_{s=0}^{r-1} M_{s \to i}, \quad M_{s \to i} \subset Q_s
\end{aligned}
\tag{4.18}
$$

Man beachte, daß die Elternmengen mehr Individuen enthalten als beim Migrationsmodell. Will man modellieren, daß Individuen in der Zielpopulation durch Kopien aus den Nachbarpopulationen ersetzt werden, so muß man von den E_i noch die entsprechende Anzahl von Individuen abziehen.

Ein Pollinationsmodell ist in Abb. 4.2 skizziert. Das zugehörige Populationsmodell lautet:

$$
\begin{aligned}
X &= \{0, \ldots, 15\} \\
\mathcal{Q} &= \Big\{\{0, \ldots, 3\}, \{4, \ldots, 7\}, \\
&\quad \{8, \ldots, 11\}, \{12, \ldots, 15\}\Big\} \\
\mathcal{E} &= \Big\{\{0, \ldots, 3, 5, 14\}, \{4, \ldots, 7, 3, 10\}, \\
&\quad \{8, \ldots, 11, 4, 15\}, \{12, \ldots, 15, 0, 9\}\Big\}
\end{aligned}
\tag{4.19}
$$

Da in diesem Modell einige Individuen von mehreren Teilpopulationen bei der Auswahl der Eltern berücksichtigt werden können, wird in gewisser Weise Information multipliziert.

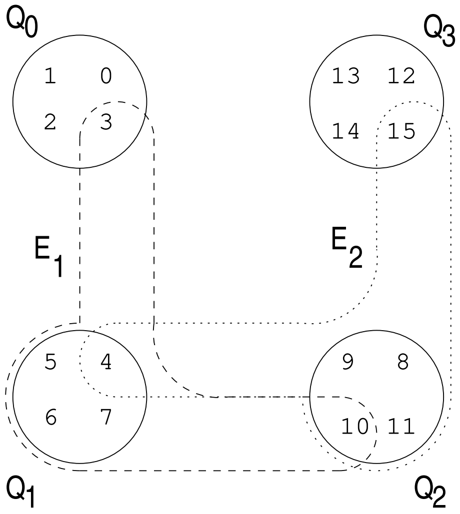

Abbildung 4.2: Schematische Darstellung des Pollinationsmodells aus Gl. 4.19 mit vier ringförmig angeordneten Teilpopulationen. Aus Gründen der Übersichtlichkeit sind nur die Hyperkanten E_1 und E_2 eingezeichnet.

4.3.3 Feinkörnige Modelle

Die überwiegende Anzahl der in der Literatur beschriebenen feinkörnig parallelen EA-Varianten verwendet als Topologie Gitter oder Tori verschiedener Dimensionen. Der Nachbarschaftsgraph des Gitters induziert dabei meist größere Nachbarschaften durch Festlegung eines maximalen Abstands von benachbarten Individuen im Gitter. So läßt sich z.B. eine Ringstruktur mit lückenloser, symmetrischer Nachbarschaft wie folgt modellieren. Zu einer Population $P = \{0, \ldots, \lambda - 1\}$ und einem Nachbarschaftsradius ρ ergibt sich

$$\begin{aligned}
\Pi_{\text{ring}} &= (X, \mathcal{E}, \mathcal{Q}) \\
X &= \{0, \ldots, \lambda - 1\} \\
\mathcal{Q} &= (\{x_0\}, \ldots, \{x_{\lambda-1}\}) \\
\mathcal{E} &= (E_0, \ldots, E_{\lambda-1}) \\
E_i &= (i - \rho, \ldots, i, \ldots, i + \rho), \quad i = 0, \ldots, \lambda - 1
\end{aligned} \tag{4.20}$$

Abb. 4.3 zeigt die Populationsstruktur eines einfachen Nachbarschaftsmodells mit $\lambda = 8$ und $\rho = 1$, also folgender Populationsstruktur:

$$\begin{aligned} X &= \{0, \ldots, 7\} \\ \mathcal{Q} &= \Big\{\{0\}, \{1\}, \{2\}, \{3\}, \\ &\quad \{4\}, \{5\}, \{6\}, \{7\}\Big\} \\ \mathcal{E} &= \Big\{\{7,0,1\}, \{0,1,2\}, \{1,2,3\}, \{2,3,4\}, \\ &\quad \{3,4,5\}, \{4,5,6\}, \{5,6,7\}, \{6,7,0\}\Big\} \end{aligned} \tag{4.21}$$

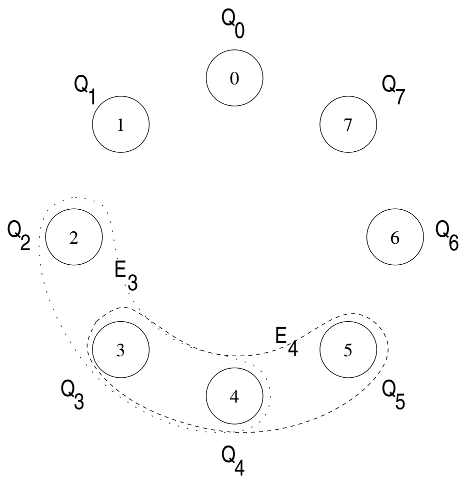

Abbildung 4.3: Schematische Darstellung eines Nachbarschaftsmodells mit acht ringförmig angeordneten Individuen. Aus Gründen der Übersichtlichkeit sind nur die Hyperkanten E_3 und E_4 eingezeichnet.

Für den in der Literatur vielfach behandelten Fall eines rechtwinkligen, zweidimensionalen Gitters bzw. Torus' bietet es sich zur Konstruktion symmetrischer, zusammenhängender Nachbarschaften an, wie bei Rudolph und Sprave (1996) als Nachbarschaft eines Individuums alle Individuen zu betrachten, die eine maximale *Champfer-Distanz* entfernt sind[3].

[3]Die Champfer-Distanz stellt eine Approximation des Euklidischen Abstands in rechtwinkligen Gittern dar. Sie findet vor allem Einsatz bei der Distanztransformation in der elektronischen Bildverarbeitung. Das zugehörige Populationsmodell wird in Abschnitt 8.3 definiert.

Die Abbildung feinkörniger EA-Populationsmodelle auf Euklidische Räume folgt meist aus biologischer Analogie (Gorges-Schleuter 1990), zum Teil auf dem Umweg über zelluläre Automaten (Sprave 1993). Tatsächlich stellen diese gitterinduzierten Ansätze nur Spezialfälle dar. Ein beliebiger, zusammenhängender Graph $G = (X, W), W \subseteq X \times X$ induziert daher das folgende feinkörnige Populationsmodell:

$$\begin{aligned}
\Pi_{\text{graph}} &= (X, \mathcal{E}, \mathcal{Q}) \\
X &= \{0, \ldots, \lambda - 1\} \\
\mathcal{Q} &= (\{x_0\}, \ldots, \{x_{\lambda-1}\}) \\
\mathcal{E} &= (E_0, \ldots, E_{\lambda-1}) \\
E_i &= \{j \mid d_G(i, j) \leq \rho\}, \quad 1 \leq \rho \leq \lambda/2,\ i, j \in X
\end{aligned} \tag{4.22}$$

4.3.4 Modellierung von Nachkommenüberschuß

Die klassischen (μ, λ)- und $(\mu + \lambda)$-Verfahren der Evolutionsstrategien modellieren einen Geburtenüberschuß innerhalb einer Population, der durch umweltbedingte Selektion wieder reduziert wird. Oft wird daher bei den Evolutionstrategien die Größe der Elternpopulation als Populationsgröße aufgefaßt. Um aber mit anderen Selektionsmethoden vergleichbar zu bleiben, ist es sinnvoll, statt dessen die Anzahl der Nachkommen bzw. bei der $(\mu + \lambda)$-Selektion die Summe aus Eltern- und Nachkommenzahl zu verwenden. Bei der Modellierung der lokalen (μ, λ)-Selektion aus (Sprave 1994) entstehen dann an jeder Position in der Elternpopulation panmiktische Teilpopulationen der Größe λ/μ und somit ein Pollinationsmodell mit kleinen Teilpopulationen. Tatsächlich kann jedes Reproduktionsverfahren in Evolutionären Algorithmen mit dem Prinzip des Nachkommenüberschusses kombiniert werden. Man erhält dann ein Verfahren mit Partnerwahl (mating selection) und Umweltselektion (environmental selection). In Kapitel 5 werden zwei verschiedene Varianten der lokalen (μ, λ)-Selektion definiert.

4.4 Modellierung von Isolationszeiten

Die bisherige Definition von Populationsstrukturen ermöglicht nicht die Modellierung von Isolationszeiten. Wegen der offensichtlichen Ähnlichkeit zu klassischen Flußproblemen bietet sich bei der Modellierung mittels einfacher Graphen die Verwendung von Kantengewichten, etwa dem Kehrwert der Isolationszeit, an.

Aber auch die oben angegebene Definition von Populationsstrukturen läßt sich leicht so erweitern, daß Isolationszeiten modellierbar werden. Anstelle einer eines Hypergraphs H verwenden wir eine Folge von Hypergraphen $H^{(t)}$, so daß die Populationsstruktur zu jeder Generation eine andere sein kann.

Def. 7 Eine *dynamische Populationsstruktur* Π auf einer Population P mit $|P| = \lambda$ ist ein Paar $(H^{(t)}, Q)$, bestehend aus einer Folge von Hypergraphen $H^{(t)} = (X, \mathcal{E}^{(t)}), X = \{\overline{0}, \dots, \overline{\lambda - 1}\}, \mathcal{E}_t \subseteq \mathcal{P}(X)$ und $\forall t \in \mathbb{N} : |\mathcal{E}^{(t)}| = D$, sowie aus einer Partition $Q \subset \mathcal{P}(\mathcal{X})$ von X mit $|Q| = D$. Die Hyperkante $E_i^{(t)}$ heißt *Deme zum Zeitpunkt t* aller Elemente aus $Q_i \in Q$. Eine dynamische Populationsstruktur mit $\mathcal{E}^{(t)} = \mathcal{E} = const$ für alle $t \in \mathbb{N}$ heißt auch *statische Populationsstruktur* oder einfach *Populationsstruktur*. Ist die Folge $H^{(t)}$ einer dynamischen Populationsstruktur zyklisch, so heißt auch die Populationsstruktur *zyklisch*. Die Länge eines Zyklus' heißt *Periode* der Populationsstruktur.

In dieser Definition ist der bisherige Begriff der Populationsstruktur als Spezialfall enthalten. Wir definieren weiterhin als Verallgemeinerung von Def. 6:

Def. 8 Der *Durchmesser* Δ_Π einer dynamischen Populationsstruktur $\Pi = (X, \mathcal{E}^{(t)}, \mathcal{Q}), |X| = \lambda$ ist definiert als

$$\Delta_\Pi \quad := \quad min \left\{ k \in \mathbb{N} \cup \{\infty\} \mid \sum_{i=1}^{k} \prod_{t=1}^{i} \overline{E^{(t)}}\, \overline{Q}^\top = 1^{\lambda \times \lambda} \right\}. \tag{4.23}$$

Eine dynamische Populationsstruktur mit $\Delta_\Pi \neq \infty$ heißt *zusammenhängend*.

Analog zu Gleichung 4.18 sei die Nachbarschaftsbeziehung zwischen den Teilpopulationen durch einen Graph G beschrieben, der die Knotenmenge $V = \{0, \dots, r-1\}$ besitzt und die Kantenmenge $E \subseteq V \times V$ mit $(s,t) \in E$, falls einen

Migrationspfad von Q_s nach Q_t existiert. Dazu gebe es zu jeder Kante $v = (s, t)$ eine ausgezeichnete Teilmenge $M_{s\to t}$ von Q_s der Migranten von Q_s nach Q_t.

Ein Pollinationsmodell mit einer Isolationsdauer von η Generationen läßt sich dann wie folgt als dynamische Populationsstruktur beschreiben:

$$\begin{aligned}
\Pi_{\text{poll}} &= (X, \mathcal{E}^{(t)}, \mathcal{Q}) \\
X &= \{0, \ldots, \lambda - 1\} \\
Q_i &= \{i\nu, \ldots, i\nu + \nu - 1\} \\
\mathcal{E}^{(t)} &= (E_0^{(t)}, \ldots, E_{r-1}^{(t)}) \\
E_i^{(t)} &= \begin{cases} Q_i \cup \bigcup_{(s,i)\in G} M_{s\to i} & \text{für } t \equiv 0 \; mod \; \eta \\ Q_i & \text{sonst} \end{cases}
\end{aligned} \tag{4.24}$$

mit $M_{s\to i} \subset Q_s$. Als weitere Verallgemeinerung kann man die Partition $\mathcal{Q}$ ebenfalls durch eine Folge von Partitionen ersetzten. Man kann dann eine intuitivere Modellierung der Migration formulieren, indem man $\mathcal{E}^{(t)} = \mathcal{Q}^{(t)}$ setzt und, jeweils nach Ablauf einer Isolationsperiode, Elemente aus $Q_i^{(t)}$ und $Q_j^{(t)}$ austauscht, wenn zwischen diesen ein Migrationspfad besteht. Allerdings führt dieser Ansatz zu extrem langen Zyklen in $\mathcal{Q}^{(t)}$, so daß bei Berechnungen faktisch für jede Iteration neue Matrizen aufgestellt werden müssen. Auch lassen sich keine zusätzlichen Strukturen beschreiben, wie der folgende Satz zeigt:

Satz 2 Jede Folge von statischen Populationsstrukturen $\Pi^{(t)} = (X, \mathcal{E}^{(t)}, \mathcal{Q}^{(t)})$ mit konstanter Populationsgröße $|X| = \lambda$ läßt sich durch eine dynamische Populationsstruktur vollständig beschreiben.

Beweis: Jedes Glied der Folge $\Pi^{(t)}$ läßt sich durch eine äquivalente Populationsstruktur $\Pi'^{(t)} = (X, \mathcal{E}'^{(t)}, \mathcal{Q}')$ ersetzten mit $Q_i'^{(t)} = \{i\}$ für alle $i = 0, \ldots, \lambda - 1$ und $E_i'^{(t)} = E_j^{(t)}$ für $i \in Q_j^{(t)}$. Damit ist $\mathcal{Q}'^{(t)} = const$ für $t \in \mathbb{N}_0^+$ und somit $\Pi' = (X, \mathcal{E}'^{(t)}, \{\{0\}, \ldots, \{\lambda - 1\}\}$ eine dynamische Populationsstruktur gemäß Def. 7 □.

Man beachte, daß man zusammenhängende, dynamische Populationsstrukturen erzeugen kann, bei welchen keine der aufeinanderfolgenden statischen Populationsstrukturen zusammenhängend ist, etwa wenn man ein zusammenhängendes

Migrationsmodell zeitlich so entzerrt, daß in jeder Generation höchstens ein Migrationspfad existiert oder maximal eine Population Migranten aussendet. Letzteres führt dann zu folgender Populationsstruktur:

$$
\begin{aligned}
\Pi_{\text{poll}} &= (X, \mathcal{E}^{(t)}, \mathcal{Q}) \\
X &= \{0, \ldots, \lambda - 1\} \\
Q_i &= \{i\nu, \ldots, i\nu + \nu - 1\}, \quad i = 0, \ldots, r-1 \\
\mathcal{E}^{(t)} &= (E_0^{(t)}, \ldots, E_{r-1}^{(t)}) \\
E_i^{(t)} &= \begin{cases} Q_i & \text{für} \quad t \not\equiv i \; mod \; r \\ Q_i \cup \bigcup_{(s,i) \in G} M_{s \to i}, \; M_{s \to i} \subset Q_s & \text{für} \quad t \equiv i \; mod \; r \end{cases}
\end{aligned}
\tag{4.25}
$$

4.5 Vom Modell zum Programm

Die Modellierung von Populationsstrukturen ergibt nur dann Sinn, wenn diese in einfacher Weise implementiert werden können. Der folgende Abschnitt soll zeigen, daß der in Kap. 4 präsentierte Formalismus eine Brücke zwischen Theorie und Anwendung darstellt. Es wird gezeigt, daß mit wenig Aufwand ein EA formuliert werden kann, der beliebige Populationsstrukturen gemäß Def. 7 als Eingabe akzeptiert. Besonders einfach gestaltet sich die Umsetzung, wenn die verwendete Programmiersprache bereits über einen Datentyp für Mengen oder Listen verfügt.

Der abrupte Bruch des Abstraktionsniveaus, den dieser Abschnitt darstellt, ist durchaus beabsichtigt: Es soll gezeigt werden, daß Modelle gemäß Kap. 4 in direkter Weise in die Praxis umgesetzt werden können, und daß sich die Modelle auch in konkreten Programmen wiedererkennen lassen. Denn während es bei panmiktischen EA durchaus möglich ist, mit der Angabe einer kleinen Zahl von Parametern und einem Literaturverweis auf eine formale Darstellung des Verfahrens Ergebnisse reproduzierbar zu dokumentieren, werden nichtpanmiktische

Varianten mangels formaler Modelle bisher nur in Form von Text oder wenig detailliertem Pseudo-Code beschrieben.

Für die meisten Experimente dieser Arbeit wurde das Software-Paket *Ease — Evolutionary Algorithms Scripting Environment* (Sprave 1999a) eingesetzt. Ease stellt eine Erweiterung der Script-Sprache Tcl (tool command language) von John Ousterhout (1994) dar. Hierdurch ist es möglich, die Performanz einer vorkompilierten Bibliothek von Methoden zur Manipulation von EA-Populationen mit der Flexibilität und dem hohen Abstraktionsgrad einer interpretierten Script-Sprache zu nutzen. Insbesondere verfügt Tcl über mächtige Funktion zur Listen- und String-Verarbeitung.

Ease ermöglicht verschiedene interne Repräsentationen von Individuen (Bit-String, reeller Vektor oder Kombination aus beidem). Im Falle einer reellwertigen Repräsentation sind auch individuelle Schrittweiten vorgesehen. Alle gängigen Mutations- und Rekombinationsoperatoren stehen für die jeweilige Repräsentation zur Verfügung.

Individuen werden in Ease nicht direkt, sondern über ihre Indizes in der Population angesprochen. Alle Ease-Kommandos akzeptieren Listen von Indizes, so daß evolutionäre Operatoren in einfacher Weise auf Teilmengen einer Population angewendet werden können. Ein Beispiel für die Erzeugung der Population `mypop` mit 32 Individuen mit 10 reellen Parametern, einer Schrittweite je Individuum, diskreter Rekombination auf den Objektvariablen und intermediärer Rekombination auf den Schrittweiten sieht dann so aus:

```
population mypop -size 32 -bits 0 -vars 10 -sigmas 1 \
                 -xreco discrete -sreco intermediate \
                 -select comma -relmu 0.125
```

Gleichzeitig wird (μ, λ)-Selektion mit $\frac{\lambda}{\mu} = 8$ eingestellt. Hierbei wird die (μ, λ)-Selektion als Mating-Selektion mit abschneidender Verteilung aufgefaßt. Eine Formulierung der klassischen ES-Umweltselektion mittels Ease wird von Sprave (1999a) beschrieben.

Will man nun die ersten 16 Individuen der Population `mypop` mutieren, so geschieht dies einfach durch

```
mypop mutate 0:15
```

Die Populationsstruktur aus Gl. 4.19, erweitert um eine Isolationszeit von einer Generation, läßt sich in Tcl in folgender Weise aufschreiben:

```
lappend Q { 0  1  2  3}
lappend Q { 4  5  6  7}
lappend Q { 8  9 10 11}
lappend Q {12 13 14 15}

lappend Qtmp {16 17 18 19}
lappend Qtmp {20 21 22 23}
lappend Qtmp {24 25 26 27}
lappend Qtmp {28 29 30 31}

lappend E(0) { 0  1  2  3}
lappend E(0) { 4  5  6  7}
lappend E(0) { 8  9 10 11}
lappend E(0) {12 13 14 15}

lappend E(1) { 0  1  2  3  5 14}
lappend E(1) { 4  5  6  7  3 10}
lappend E(1) { 8  9 10 11  4 15}
lappend E(1) {12 13 14 15  0  9}

set cycle 2
```

Damit werden vier Listen, `Q`, `Qtmp`, `E(0)` und `E(1)`, aufgebaut. Die Variable `cycle` gibt die Zykluslänge der Populationsstruktur an. Die Liste `Qtmp` wird als Zwischenspeicher benötigt, um die simultane Neuberechnung einer Generation zu gewährleisten. Natürlich wird man größere Populationsstrukturen wieder durch ein Programm generieren lassen. Auch hierzu stellt Ease Funktionen bereit.

Das folgende Ease-Programm stellt bis auf das Laden der Zielfunktion und die Erzeugung der Population `mypop` eine vollständige Implementierung eines EA mit beliebigen, dynamischen Populationsstrukturen gemäß Def. 7 dar:

```
set gen 0
set calls 0

while {$calls <= $maxcalls} {

    set t [expr $gen % $cycle]
    foreach offspring $Qtmp deme $E($t) {
        mypop reproduce $offspring $deme
        mypop mutate $offspring
        incr calls [mypop evaluate $offspring]
    }
    foreach newparents $Q offspring $Qtmp {
        mypop copy $newparents $offspring
    }
    set stat [mypop stat $parents]
    puts stdout "$gen $calls $stat"
    incr gen
}
```

In der ersten inneren Schleife durchlaufen die Listenvariablen `offspring` und `deme` simultan die korrespondierenden Elemente der Listen `Q` und `E($t)`. Anschließend müssen die temporär erzeugten Nachkommen in die eigentlich Population zurückgeschrieben werden. Im Falle eines panmiktischen EA werden die beiden inneren Schleifen nur einmal durchlaufen. Läßt man diese wegfallen, so erhält man die Tcl/Ease-Variante des Pseudo-Codes aus Abschnitt 2.3.1 in der **S–R–M**-Variante, da der `reproduce`-Operator in Ease Selektion und Rekombination zusammenzieht.

Unter Verwendung einer derartigen Implementierung reicht es aus, neben der auch bei panmiktischen Ansätzen notwendigen Angabe der internen Repräsentation und den verwendeten Operatoren zusätzlich das Populationsmodell anzugeben, um Ergebnisse reproduzierbar zu dokumentieren.

Kapitel 5

Selektion in Populationsstrukturen

5.1 Einleitung

Der Übergang von panmiktischen zu nichtpanmiktischen Populationsstrukturen in Evolutionären Algorithmen, sei es mit Ziel der Parallelisierung oder der Diversitätserhaltung, erfordert eine Lokalisierung des Selektionsoperators. Während wir uns in den vorangegangenen Kapiteln vor allem für die Menge der potentiellen Eltern eines Individuums interessiert haben, werden wir nun untersuchen, wie die tatsächlichen Eltern aus dieser Menge selektiert werden. Es ist nicht verwunderlich, daß die meisten lokalen Selektionsoperatoren Varianten der klassischen panmiktischen Methoden sind.

Ein traditioneller Unterschied zwischen GA und ES ist der Zeitpunkt, zu dem selektiert wird. Neuere Definitionen, z.B. von Schwefel und Kursawe (1998), von Evolutionären Algorithmen verallgemeinern dies, indem sie einen zweiten Selektionszeitpunkt in das EA-Iterationsschema einfügen. Dabei wird die Selektion zu Beginn einer Generation *Mating Selection* genannt, weil sie am ehesten der Partnerwahl in der Natur entspricht, während der Selektionsschritt am Ende einer Generation das Sterben schlecht angepaßter Individuen modelliert und daher *Environmental Selection* genannt wird. Tatsächlich ist diese Konstruktion zur Modellierung der Basisalgorithmen nicht notwendig, da beide Selektionsoperatoren in der Generationenabfolge direkt nacheinander ausgeführt werden und daher zusammengefaßt werden können. Die zweifache Selektion erleichtert aber die einheitliche Beschreibung der Basisalgorithmen und fördert damit die Akzeptanz des verallgemeinerten Iterationsschemas.

Während bei der in Kapitel 4 vorgenommenen Modellierung erst einmal die potentiellen Eltern ohne Rücksicht auf konkrete Selektionsoperatoren behandelt werden, werden nun zusätzlich die Selektionswahrscheinlichkeiten untersucht. Unter dem Gesichtspunkt der Herauszögerung der Stagnation des Optimierungsprozesses werden wir uns besonders für die Fortpflanzungswahrscheinlichkeit guter Individuen interessieren und versuchen, daraus ein Maß für den Selektionsdruck abzuleiten, der verschiedene Populationsstrukturen in Verbindung mit konkreten Selektionsoperatoren vergleichbar macht. Selektion wird häufig als ein Operator definiert, der aus einer gegebenen Population heraus eine neue gleicher Größe generiert, was bei nichtpanmiktischen Populationsstrukturen in der Regel nicht der Fall ist. Daher wird unter Selektion in dieser Arbeit das einmalige Auswählen eines Individuums aus einer gegebenen Menge von Individuen verstanden:

Def. 9 Sei M eine vollständig geordnete Menge. Ein *rangbasierter Selektionsoperator* $Sel \in M \times [0,1]$ auf M ist eine Wahrscheinlichkeitsverteilung auf $[0,\ldots,|M|-1]$.

Diese Definition vernachlässigt scheinbar den Fall, daß zwei oder mehr Individuen denselben Zielfunktionswert haben. Da aber alle in der Literatur veröffentlichten rangbasierten Selektionsverfahren eine Reihenfolge festlegen (müssen), kann man eine vollständige Ordnung als gegeben annehmen.

5.2 Panmiktische Selektionsoperatoren

Die folgende Zusammenfassung gibt einen Überblick über häufig eingesetzte panmiktische Selektionsoperatoren.

5.2.1 Proportionale Selektion

Die ursprüngliche Form (Holland 1975) der proportionalen Selektion wird bereits in Abschnitt 2.3.2 beschrieben. Da es sich dabei nicht um ein rangbasiertes

Verfahren handelt, ist die Verteilungsfunktion vom aktuellen Zustand der Population abhängig und nicht gemäß Def. 9 beschreibbar. Die wesentlichen Nachteile dieser Selektionsmethode, nämlich die Überrepräsentation eines einzelnen Superindividuums bei der Nachkommenerzeugung sowie der durch geringe Fitneßunterschiede bedingte nachlassende Selektionsdruck in der Nähe eines Optimums haben dazu geführt, daß diese Selektionsmethode nur noch von geringer praktischer Relevanz ist.

5.2.2 Rangselektion

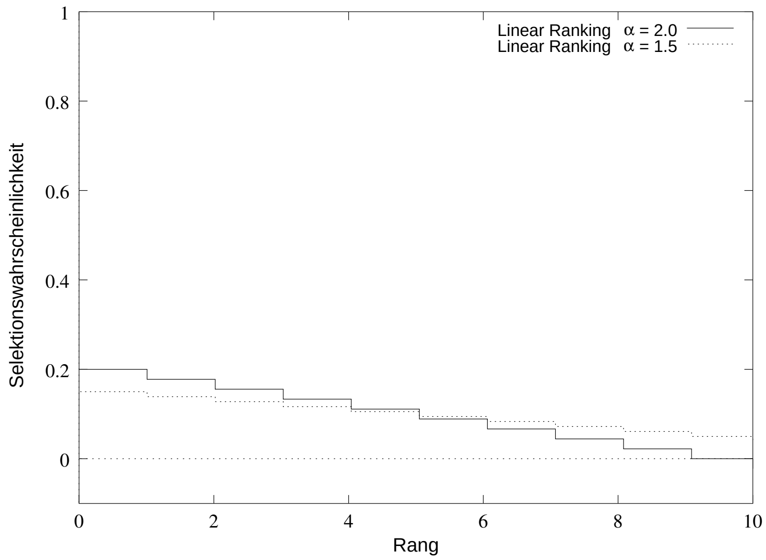

Abbildung 5.1: Selektionswahrscheinlichkeit von linearer Rangselektion für $\alpha = 1.5$ und $\alpha = 2.0$ bei einer Populationsgröße von $\lambda = 10$.

Während noch im Handbook of Evolutionary Computation (Bäck, Fogel und Michalewicz 1997) proportionale Selektion als wesentliches Merkmal eines Genetischen Algorithmus' bezeichnet wird (Eshelman 1997), haben die Nachteile dieser Selektionsmethode zur Entwicklung von GA auf der Basis rangbasierter Selektionsoperatoren geführt. Bereits Baker (1985) schlägt eine lineare Abhängigkeit (Linear-Ranking-Selektion) zwischen dem Rang eines Individuums und seiner Selektionswahrscheinlichkeit vor. Dabei hängt bei gegebener Populationsgröße λ die Selektionsverteilung von genau einem Parameter ab, nämlich dem Erwartungswert für die Anzahl der Nachkommen des besten Individuums. Sei 0 der Rang des besten Individuums und $\lambda - 1$ der Rang des schlechtesten Individuums, dann ist die Selektionswahrscheinlichkeit für das Individuum mit Rang i

$$p_{linrank}(i) = \frac{\alpha + [i/(\lambda - 1)](\beta - \alpha)}{\lambda} \tag{5.1}$$

mit $1 \leq \beta \leq \alpha \leq 2$ und $\alpha + \beta = 2$, so daß sich die lineare Rangselektion über einen Parameter steuern läßt.

5.2.3 Turnierselektion

Auch Turnierselektion kann als rangbasierte Selektion gemäß Def. 9 beschrieben werden, obwohl die Verteilung nur implizit durch die Turnierregeln gegeben ist. Es soll hier nur die klassische Variante der deterministischen Turniere mit Zurücklegen betrachtet werden. Es wird also für jede Position in der Population ein Turnier der Größe k durch gleichverteiltes Ziehen aus allen λ Individuen zusammengestellt. Das beste dieser k Individuen wird deterministisch übernommen. Dann ist die Wahrscheinlichkeit, daß in einem einzelnen Turnier das Individuum mit Rang i aus einer Population von λ Individuen und Turniergröße k bei einem Ziehen selektiert wird, nach Bäck (1994)

$$p_{tournament}(i) = \left(1 - \frac{i}{\lambda}\right)^k \left[1 - \left(\frac{\lambda - i - 1}{\lambda - i}\right)^k\right] . \tag{5.2}$$

Wie auch in Abb. 5.2 zu sehen ist, wird die Turnierselektion für eine Turniergröße von $k = 2$ linear. Durch Koeffizientenvergleich erhält man $\alpha = 2 - 1/\lambda$ als entsprechendem Parameter beim Linear Ranking sowie $\alpha = 2$ für Turnierselektion ohne Zurücklegen. Für Turniergrößen $k > 2$ entsteht eine nichtlineare Rangselektion mit einem Polynom $k - 1$-ten Grades. Man kann also Turnierselektion auch als effiziente Implementierung von polynomieller Rangselektion auffassen.

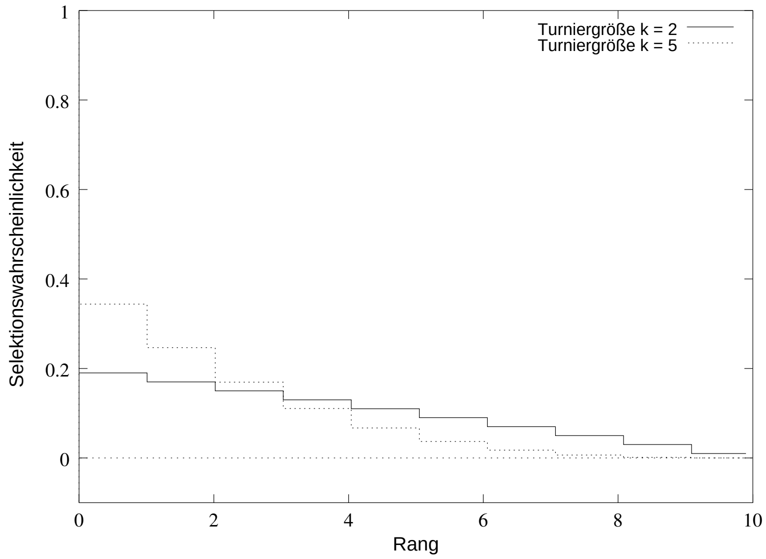

Abbildung 5.2: Selektionswahrscheinlichkeit von Turnierselektion mit Turniergrößen $k = 2$ und $k = 5$ und Populationsgröße $\lambda = 10$.

5.2.4 $(\mu \overset{+}{,} \lambda)$-Selektion

In der klassischen Evolutionsstrategie wird Selektion als Reduktion eines Geburtenüberschusses verstanden und daher oft als Umweltselektion bezeichnet. Genauso gut kann man aber die $(\mu \overset{+}{,} \lambda)$-Selektion als Partnerselektion (mating selection) mit einer rechteckigen Verteilungsfunktion auffassen, wie es in (Hoffmeister und Bäck 1990) formalisiert wurde. Zur Unterscheidung wird dann auch oft von Truncation-Selektion gesprochen. Die Selektionswahrscheinlichkeit bei einmaligem Ziehen für ein Individuum mit Rang i beträgt dann bei der Kommaselektion

$$p_{comma}(i) \;=\; \begin{cases} 1/\mu & : \; i < \mu \\ 0 & : \; \text{sonst.} \end{cases} \tag{5.3}$$

Man erhält dann eine Populationsgröße von λ, während μ als Anzahl der Elternindividuen mit positiver Selektionswahrscheinlichkeit zum Parameter des Selektionsoperators wird. Die Selektionswahrscheinlichkeit einer (μ, λ)-Selektion über

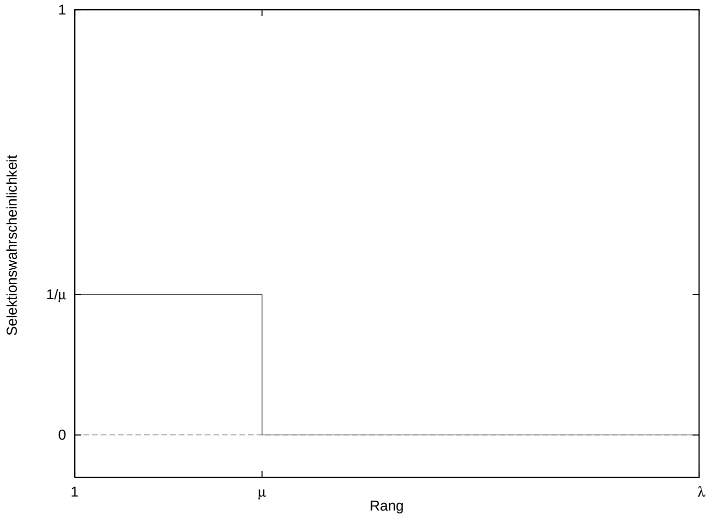

Abbildung 5.3: Selektionswahrscheinlichkeit einer $(4, 10)$-Selektion über die Ränge.

die Ränge ist in Abb. 5.3 dargestellt. Verallgemeinerungen dieser Selektionsvariante wurden in jüngster Zeit von Schönemann (1999) untersucht. Dabei wurde die Gleichverteilung beim Ziehen aus den μ besten Individuen durch nicht monotone Verteilungen ersetzt.

5.3 Lokale Selektion

Es stellt sich nun die Frage, wie die panmiktischen Selektionsoperatoren modifiziert werden müssen, um auf die in Kapitel 4 definierten Mengen potentieller Eltern eingeschränkt zu werden, also auf die Hyperkanten aus $\mathcal{E}$ einer Populationsstruktur $\Pi = (X, \mathcal{E}, \mathcal{Q})$.

5.3.1 Lokale proportionale Selektion

Zu einer gegebenen Populationsstruktur $\Pi = (X, \mathcal{E}, \mathcal{Q})$ erhält man als *proportionalen Selektionsoperator* für eine beliebige Hyperkante $E \in \mathcal{E}$ die lokale relative

Fitneß p_E mit:

$$p_E(x,t) \;:=\; \frac{F(\hat{x}_t)}{\sum\limits_{y \in E} F(\hat{y}_t)} \,, \tag{5.4}$$

wobei $F : M \to R^+, M \subseteq \mathbb{R}^n$ die Fitneßfunktion zu einem gegebenen Optimierproblem und $\hat{x}_t$ die interne Repräsentation des Individuums mit dem Index x zum Zeitpunkt t ist. Analog zur panmiktischen proportionalen Selektion kann nun die lokale kumulative Dichtefunktion (CDF) ermittelt werden als

$$CDF_E(x,t) \;:=\; \sum_{y \in E : y < x} p_E(x,t) \tag{5.5}$$

und durch gleichverteiltes Ziehen von ξ aus $[0,1)$ wählt man das Individuum k mit

$$CDF_E(k,t) \;=\; min\,\{i \in E : CDF_E(i,t) \geq \xi\} \;. \tag{5.6}$$

5.3.2 Lokale Rangselektion

Zur Lokalisierung der Rangselektion müssen die Individuen innerhalb eines Dems $E_j \in \mathcal{E}$ nach ihrer Fitneß sortiert werden. Weist man nun jeweils dem Rang i die relative Fitneß $p_{linrank}(i)$ aus Gl. 5.1 zu, wobei man λ durch $|E_j|$ ersetzt, so kann man wiederum gemäß Gl. 5.5 und 5.6 mittels Roulette-Wheel-Selektion ziehen.

5.3.3 Lokale Turnierselektion

Turnierselektion läßt sich in trivialer Weise lokalisieren, indem die Turnierteilnehmer für Eltern von Individuen aus $Q_j \in \mathcal{Q}$ gleichverteilt aus E_j gezogen werden.

5.3.4 Lokale (μ, λ)-Selektion

Bei der lokalen (μ, λ)-Selektion wird das Verhältnis λ/μ zum Parameter des lokalen Selektionsoperators. Der Überschuß, der in der panmiktischen ES generiert wird, wird hier durch die Elternmengen als gegeben angenommen, so daß die Eltern einer Teilpopulation Q_j gleichverteilt aus den besten $\lfloor\, |E_j|\, \mu/\lambda\, \rfloor$ Individuen aus E_j gezogen werden. Für $|E_j| = \lambda/\mu$ wird dann immer der beste potentielle Elter ausgewählt, kleinere Werte von $|E_j|$ ergeben keinen Sinn.

Die von Rudolph (1990) verwendete lokale Kommaselektion zusammen mit einem Migrationsmodell läßt sich im wesentlichen auf diese Weise modellieren. Lediglich die Auswahl der Migranten weicht ab, da zur Migration nun alle Nachkommen ausgewählt werden können.

Ein alternativer Ansatz zur feinkörnigen Kommaselektion (und auch Plusselektion), bei dem der Nachkommenüberschuß explizit erzeugt wird, wurde von Sprave (1994) vorgestellt. Das Verfahren ist zweistufig: Zunächst werden für jede Position in der Population λ/μ Nachkommen erzeugt, deren Eltern gleichverteilt aus einer gegebenen Nachbarschaft gezogen wurden. Anschließend wird, wie in Abb. 5.4 exemplarisch dargestellt, der jeweils beste, lokal erzeugte Nachkomme übernommen. Im Fall der Plusselektion wird das ursprüngliche Individuum an einer Position beibehalten, wenn es besser als alle lokalen Nachkommen ist.

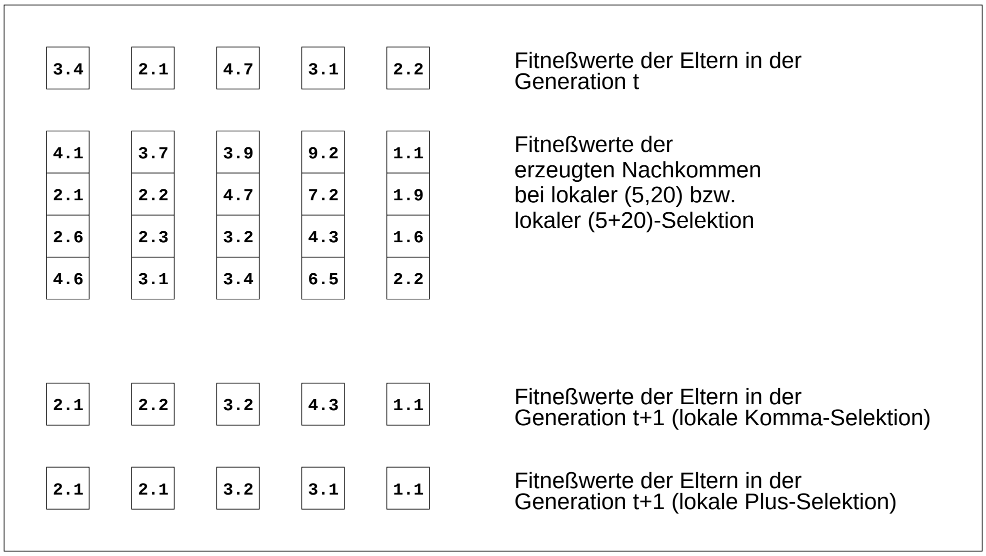

Abbildung 5.4: Lokale (μ, λ)- und $(\mu + \lambda)$-Selektion nach Sprave (1994).

5.3.5 Spezielle lokale Selektionsoperatoren

Eine Variante der lokalen Turnierselektion auf Gittertopologien ist die sogenannte Random-Walk-Selektion (Gordon 1994; Sarma 1998): Ausgehend vom zu ersetzenden Individuum wird ein kurzer Random-Walk entlang des Gitters durchgeführt. Selektiert wird das beste besuchte Individuum. Die Idee dieser Selektionsmethode ist es, nahegelegene Individuen mit höherer Wahrscheinlichkeit zu selektieren als weiter entfernte.

Häufig wird lokale Selektion auch auf der Basis einer detaillierteren Modellierung natürlicher Selektionsprozesse definiert. Hierzu bieten sich Räuber-Beute-Modelle an, wie sie z.B. von Mehnen (1994) untersucht wurden. Die wesentliche Idee dabei ist, daß sich Räuber- und Beuteindividuen auf einer gemeinsamen Topologie (Gitter, Torus, Graph) bewegen können. Schwache Beuteindividuen fallen dabei eher den Räubern zum Opfer als starke. Bei Laumanns, Rudolph und Schwefel (1998) wird mit dem Ziel der mehrkriteriellen Optimierung noch zwischen den Räubern unterschieden: Jeder Räuber selektiert nach nur einem Kriterium, und zu jeder Zielfunktion gibt es mindestens einen Räuber. Die Räuber bewegen sich zufällig (random walk) auf einem zweidimensionalen Torus, auf welchem die Beuteindividuen plaziert sind.

Kapitel 6

Takeover-Verhalten von Populationsmodellen

Die in Kap. 4 vorgenommene Modellierung soll nicht nur als Formalismus zur exakten Beschreibung nichtpanmiktischer EA dienen, sondern auch als Basis für theoretische Analysen. Um gewissermaßen die Theorietauglichkeit des vorgestellten Modells aufzuzeigen, wird in diesem Kapitel eine verbreitete Technik zur Untersuchung von Selektionsoperatoren in EA, die Analyse des Takeover-Verhaltens, auf Populationsstrukturen gemäß Def. 7 übertragen.

6.1 Der Begriff der Takeover-Zeit

Ein wesentliches Problem bei Evolutionären Algorithmen ist das mögliche Zusammenfallen der gesamten Population auf ein kleines Intervall des Suchraums, im diskreten Fall sogar auf einen einzigen Punkt, bevor das Optimum hinreichend angenähert oder erreicht ist. Zur Analyse der vorzeitigen Stagnation wurde der Begriff der Takeover-Zeit (Goldberg und Deb 1991) eingeführt, um ein Maß für die Neigung eines Selektionsoperators zur vorzeitigen Stagnation zu bekommen. Zur Ermittlung der Takeover-Zeit wird die Anzahl der Generationen, die ein einzelnes, bestes Individuum in der Initialpopulation benötigt, um sich unter alleiniger Anwendung des Selektionsoperators auf die gesamte Population auszubreiten, gemessen, berechnet oder geschätzt.

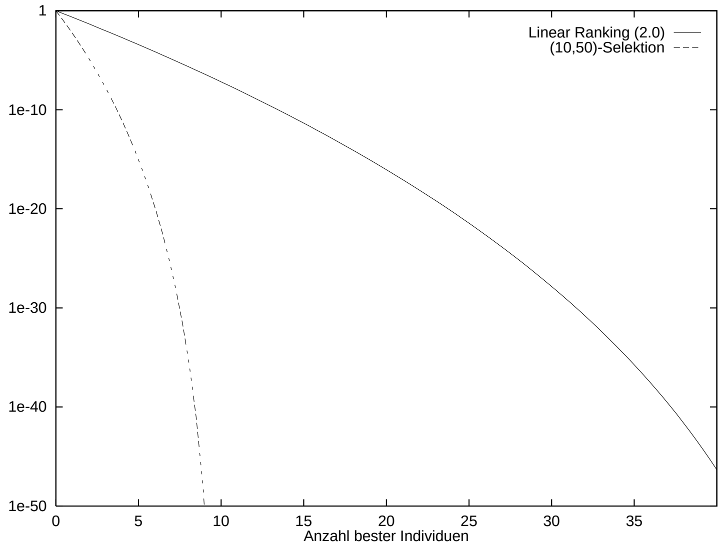

Abbildung 6.1: Wahrscheinlichkeit des Aussterbens aller besten Individuen innerhalb einer Generation in Abhängigkeit von der Anzahl der besten Individuen.

Hierbei wird sofort ein Problem dieses Begriffs sichtbar. Bei nichtelitärer Selektion wird das beste Individuum in jeder Generation mit positiver Wahrscheinlichkeit nicht selektiert, also vergessen. Da mit einem einzelnen besten Individuum gestartet wird, ist diese Wahrscheinlichkeit zu Beginn sehr hoch, nimmt dann jedoch deutlich ab. Einen Ausweg aus diesem Problem bietet die Definition der *Takeover-Wahrscheinlichkeit* von Chakraborty, Deb und Chakraborty (1996). Hierzu wird ein stochastischer Prozeß $\{G^{(t)}\}$ verwendet, der die Anzahl der besten Individuen in der t-ten Generation modelliert. Der Zustandsraum des Prozesses ist $\{0, \ldots, \lambda\}$ für eine Population mit λ Individuen. Da die Anzahl der Individuen in der $t + 1$-sten Generation nur vom Zustand zum Zeitpunkt t abhängt, bildet der Prozeß eine Markov-Kette. Um zu berechnen, wie hoch die Wahrscheinlichkeit des Zustands λ nach t Generationen ist, benötigt man alle Übergangswahrscheinlichkeiten $p_{ij}, i, j \in \{0, \ldots, \lambda\}$. Die Autoren bestimmen hierzu für eine Reihe von Selektionsoperatoren die Wahrscheinlichkeit $p_{select}(\lambda, k)$, mit der man durch einmaliges Ziehen aus einer Population, in der genau k von λ Individuen beste Individuen sind, ein bestes Individuum erhält. Es handelt sich also um k Bernoulli-Experimente mit konstanter Erfolgswahrscheinlichkeit, so daß die Anzahl der Erfolge binomialverteilt ist. Die Übergangswahrscheinlichkeit zwischen nichtabsorbierenden Zuständen $i, j \in \{1, \ldots, \lambda - 1\}$ ist dann

$$p_{i,j} = \binom{\lambda}{j} \Big(p_{select}(i, \lambda)\Big)^j \Big(1 - p_{select}(i, \lambda)\Big)^{\lambda - j}. \tag{6.1}$$

Es stellt sich nun die Frage, ob sich dieser Ansatz auf nichtpanmiktische Populationsmodelle übertragen läßt. Eine wesentliche Eigenschaft, die durch den Übergang auf beliebige Populationsmodelle verloren geht, ist die Ununterscheidbarkeit der Individuen. Der Zustandsraum Ξ eines geeigneten stochastischen Prozesses muß daher jede Position in der Population berücksichtigen, also

$$\Xi = \mathbb{B}^\lambda \tag{6.2}$$
$$\xi = (\xi_0, \ldots, \xi_{\lambda-1})^\mathsf{T} \in \Xi \tag{6.3}$$
$$\xi_i = \begin{cases} 1 & : \text{ an Position } i \text{ ist ein bestes Individuum} \\ 0 & : \text{ sonst.} \end{cases} \tag{6.4}$$

Zu einer gegebenen Populationsstruktur $\Pi = (X, \mathcal{E}, \mathcal{Q})$ sei

$$\#_\xi E_j = \sum_{\nu \in E_j} \xi_\nu, \; j = 0, \ldots, |\mathcal{E}| - 1 \tag{6.5}$$

die Anzahl bester Individuen in E_j im Zustand $\xi \in \Xi$. Die Übergangswahrscheinlichkeit $p_{\xi,\sigma}$ von einem Zustand $\xi \in \Xi$ nach $\sigma \in \Xi$ ist dann

$$p_{\xi,\sigma} = \prod_{i=0}^{\lambda-1} [\sigma_i p_i + (1 - \sigma_i)(1 - p_i)] \text{ mit} \tag{6.6}$$
$$p_i = p_{select}(|E_j|, \#_\xi E_j) \text{ für } i \in Q_j. \tag{6.7}$$

Aufgrund des mit der Anzahl der Individuen exponentiell wachsenden Zustandsraums ist das Ausrechnen der Takeover-Wahrscheinlichkeit für beliebige Populationsmodelle gemäß Chakraborty, Deb und Chakraborty (1996) praktisch unmöglich. Allerdings wurde bisher auch nicht ausgenutzt, daß die p_i nicht notwendigerweise verschieden sind. Da innerhalb einer Teilpopulation Q_i jede Konfiguration mit einer gegebenen Anzahl bester Individuen gleich ist, kann man

Gl. 6.6 daher auch folgendermaßen schreiben:

$$
\begin{aligned}
p_{\xi,\sigma} &= \prod_{i=0}^{|\mathcal{Q}|-1} p_i^{\#_\sigma Q_i}(1-p_i)^{|Q_i|-\#_\sigma Q_i} \text{ mit} && (6.8)\\
p_i &= p_{select}(|E_i|, \#_\xi E_i). && (6.9)
\end{aligned}
$$

Zwar erreicht man hierdurch bei den Q_i den Übergang von den exakten Posi-

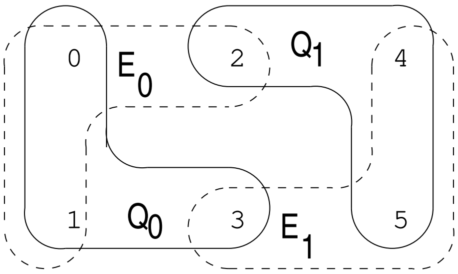

Abbildung 6.2: Migrationsmodell mit zwei Teilpopulationen und disjunkten Migrantenmengen.

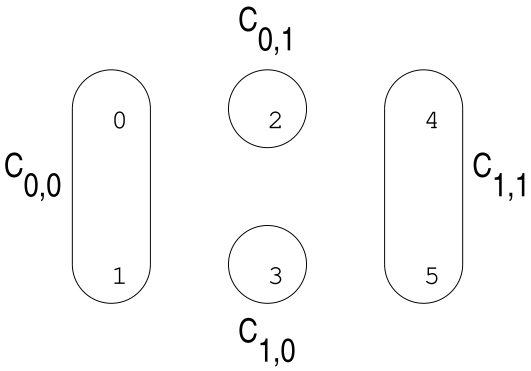

Abbildung 6.3: Partition der Populationsstruktur in Abb. 6.2 gemäß Gl. 6.10.

tionen zu den Anzahlen bester Individuen, dies führt jedoch nicht zu einer Verkleinerung des Zustandsraums, da in p_i ja weiterhin die E_i enthalten sind. Etwas einfacher ist der Fall, wenn $\mathcal{E}$ eine Partition von X bildet, wie z.B. in Gl. 4.17 unter der Annahme disjunkter Migrationsmengen $M_{s \to t}$. Man erhält dann eine Partition $\mathcal{C}$ von X mittels

$$C_{i,j} = E_i \cap Q_j,\ i,j \in \{0, \ldots, |\mathcal{E}| - 1\}, \tag{6.10}$$

da $\mathcal{E}$ und $\mathcal{Q}$ bereits Partitionen von X sind und somit jedes $x \in X$ in genau einem $E_i \in \mathcal{E}$ und genau einem $Q_j \in \mathcal{Q}$ ist, also genau in $E_i \cap Q_j$. Die $C_{i,j}$ sind genau diejenigen Teilmengen der E_i, die dieselbe Wahrscheinlichkeit bester Individuen aufweisen. Man kann nun also den Zustandsraum Ξ definieren als

$$\Xi \subseteq \mathbb{N}^{|\mathcal{C}|} \times \mathbb{N}^{|\mathcal{C}|} \tag{6.11}$$

$$\xi = (\xi_{0,0}, \ldots, \xi_{|\mathcal{C}|-1,|\mathcal{C}|-1})^{\mathsf{T}},\ 0 \leq \xi_{i,j} \leq |C_{i,j}| \tag{6.12}$$

$$\xi_{i,j} = k \ \Leftrightarrow C_{i,j} \text{ enthält } k \text{ beste Individuen.} \tag{6.13}$$

Die Mächtigkeit des Zustandsraums ist gleich dem Produkt der Zustände der nichtleeren $C_{i,j}$:

$$|\Xi| = \prod_{(i,j) \in \{0,\ldots,|\mathcal{C}|-1\}^2} (|C_{i,j}| + 1). \tag{6.14}$$

Die Übergangswahrscheinlichkeit $p_{\xi,\sigma}$ von einem Zustand $\xi \in \Xi$ nach $\sigma \in \Xi$ ist nun

$$p_{\xi,\sigma} = \prod_{(i,j) \in \{0,\ldots,|\mathcal{C}|-1\}^2} \binom{|C_{i,j}|}{\sigma_{i,j}} p_j^{\sigma_{i,j}} (1 - p_j)^{|C_{i,j}| - \sigma_{i,j}} \tag{6.15}$$

$$p_j = p_{select}(|E_j|, \#_\xi E_j). \tag{6.16}$$

Bei einem Migrationsmodell mit ringförmiger Anordnung der Teilpopulationen hat jede Teilpopulation drei Nachbarn (einschließlich ihrer selbst), so daß die Anzahl der nichtleeren $C_{i,j}$ gerade $3|\mathcal{Q}|$ ist. Bei vier Teilpopulationen der Größe 10 und jeweils nur einem Migranten zwischen benachbarten Teilpopulationen, also $|\mathcal{Q}| = |\mathcal{E}| = 4, |Q_i| = |E_i| = 10$ und $|C_{i,j}| \in \{0, 1, 8\}$, erhält man folgende

Populationsstruktur:

$$\begin{aligned} X &= \{0,\ldots,39\} & & \\ Q_0 &= \{0,\ldots,9\} & E_0 &= \{10,1,\ldots,8,39\} \\ Q_1 &= \{10,\ldots,19\} & E_1 &= \{20,11,\ldots,18,9\} \\ Q_2 &= \{20,\ldots,29\} & E_2 &= \{30,21,\ldots,28,19\} \\ Q_3 &= \{30,\ldots,39\} & E_3 &= \{0,31,\ldots,38,29\} \end{aligned} \tag{6.17}$$

Die Partition $\mathcal{C}$ kann dann als

$$C_{i,j} = \begin{cases} \{10i+1,\ldots,10i+8\} & : \ i=j \\ \{10i\} & : \ i=j-1 \\ \{10i-1\} & : \ i=j+1 \\ \emptyset & : \ \text{sonst} \end{cases} \tag{6.18}$$

$$i,j \in \{0,1,2,3\} \tag{6.19}$$

geschrieben werden, wobei alle Indexrechnungen in der Restklassengruppe modulo 40 durchgeführt werden. Da also 4 Mengen mit je 9 möglichen Zuständen und 8 Mengen mit 2 Zuständen existieren, ist der Zustandsraum

$$\Xi = \{0,\ldots,8\}^4 \times \{0,1\}^8, \tag{6.20}$$

so daß man gemäß Gl. 6.14 immer noch beachtliche $|\Xi| = 9^4 2^8 = 1.679.616$ Zustände erhält. Für die Berechnung der Takeover-Wahrscheinlichkeit bestimmt den Aufwand aber weniger die Anzahl der Zustände als die Anzahl der Zustandsübergänge mit positiver Wahrscheinlichkeit. Da neben dem Verlassen der absorbierenden Zustände $\sigma_{i,j} = 0$ für alle $i,j \in \{0,\ldots,|\mathcal{E}|-1\}$ und $\sigma_{i,j} = |C_{i,j}|$ für alle $i,j \in \{0,\ldots,|\mathcal{E}|-1\}$ nur wenige Zustandsübergänge die Wahrscheinlichkeit 0 aufweisen, erhält man jedoch keine dünnbesetzte Matrix.

Etwas anders sieht es aus, wenn man Individuen nur durch bessere ersetzt, wie etwa bei der $(\mu + \lambda)$-Selektion. Hier weisen nur diejenigen Zustandsübergänge eine von 0 verschiedene Übergangswahrscheinlichkeit auf, bei welcher die Anzahl bester Individuen in jeder Teilmenge der Partition nicht kleiner wird, also gilt wegen $k < 0 \Rightarrow \binom{n}{k} = 0$

$$p_{\xi,\sigma} = \prod_{(i,j)\in\{0,\ldots,|\mathcal{C}|-1\}^2} \binom{|C_{i,j}|}{\sigma_{i,j} - \xi_{i,j}} p_j^{\sigma_{i,j}-\xi_{i,j}} (1-p_j)^{|C_{i,j}|-\sigma_{i,j}} \tag{6.21}$$

$$p_j = p_{select}(|E_j|, \#_\xi E_j). \tag{6.22}$$

Im Beispiel aus Gl. 6.17 erhält man dann wegen

$$|\{(\xi,\sigma) \in \Xi^2 : p_{\xi,\sigma} > 0\}| = \prod_{C_{i,j}\neq\emptyset} \frac{1}{2}|C_{i,j}|(|C_{i,j}|+1) \tag{6.23}$$

immer noch ca. $1,1 \cdot 10^{10}$ Zustandsübergänge mit positiver Wahrscheinlichkeit.

6.2 Probabilistischer Durchmesser

Wenn schon eine exakte Modellierung als Markov-Prozeß an der Größe des Zustandsraums scheitert, so kann man dennoch versuchen, zu einer sinnvollen Abschätzung zu gelangen, ähnlich wie dies für panmiktische EA in der Takeover-Time von Goldberg und Deb (1991) durchgeführt wurde.

In Def. 6 wurde der Durchmesser einer Populationsstruktur eingeführt. Da dieser gemäß Satz 1 gleichzeitig die Elitist-Takeover-Time einer Populationsstruktur liefert, bietet es sich an, ein ähnliches Maß für nichtelitäre Selektion zu definieren. Die Idee dahinter ist, ein bestes Individuum unter Anwendung eines gegebenen Selektionsoperators durch eine Populationsstruktur zu propagieren, bis jedes Individuum mit hoher Wahrscheinlichkeit erreicht wurde.

Startet man mit einem einzelnen besten Individuum in der ersten Generation, so ist der Erwartungswert für die Anzahl bester Individuen in der Folgegeneration

$$\mathsf{E}_{best}^{(1)} := \lambda \cdot p_{select}(\lambda, 1). \tag{6.24}$$

Allgemein erhält man für eine gegebene Anzahl $k^{(t)}$ bester Individuen in Generation t als Erwartungswert für die Anzahl bester Individuen in Generation $t+1$

$$\mathsf{E}_{best}^{(t+1)} := \lambda \cdot p_{select}(\lambda, k^{(t)}). \tag{6.25}$$

Die Bildung des Erwartungswertes kann man nun iterieren und in Gl. 6.24 einsetzen. Da eine Verteilung im allgemeinen nicht vollständig durch ihren Erwartungswert definiert ist, kann man nicht davon ausgehen, daß man durch t-fache Iteration tatsächlich den Erwartungswert für die Anzahl bester Individuen nach t Generation erhält. Setzt man dennoch

$$k^{(t+1)} := \mathsf{E}^t_{best} \tag{6.26}$$

so erhält man eine Annäherung, wobei durch die wiederholte Bildung des Erwartungswertes die Möglichkeit der Auslöschung aller besten Individuen herausgemittelt wird. Tatsächlich wird diese Vereinfachung auch bei der Ermittlung der Takeover-Zeit für Evolutionäre Algorithmen getan, so auch in der als Begriffsdefinition geltenden Veröffentlichung von Goldberg und Deb (1991, S. 71, Gl. 4). Die so ermittelten Takeover-Zeiten haben sich durchaus als Werkzeug zur Bewertung von Selektionsoperatoren bewährt. Die folgende Definition verzichtet daher auf die explizite Erwähnung des Erwartungswerts:

Def. 10 Sei $\Pi = (X, \mathcal{E}^{(t)}, \mathcal{Q}), |X| = \lambda$ eine Populationsstruktur, und $p_{select} : \mathbb{N} \times \mathbb{N} \to \mathbb{R}, (\lambda, k) \mapsto p_{select}(\lambda, k)$ die Erfolgswahrscheinlichkeit eines gegebenen Selektionsoperators in Abhängigkeit von k und λ. Weiterhin sei für alle $i \in X$:

$$s_i^{(1)} = 1/\lambda \tag{6.27}$$

$$r_i^{(t)} = \sum_{j \in E_\nu^{(t)}} s_j^{(t)}, \quad i \in Q_\nu \tag{6.28}$$

$$s_i^{(t+1)} = p_{select}(r_i^{(t)}, |E_\nu^{(t)}|), \quad i \in Q_\nu \tag{6.29}$$

Der *probabilistische Durchmesser* von Π unter p_{select} ist

$$\Delta_\varepsilon(\Pi, p_{select}) := min\{t : \forall i \in X : s_i^{(t)} \geq 1 - \varepsilon\} \tag{6.30}$$

Der Wert $1 - \varepsilon$ ist der gewünschte Takeover-Level, wobei ε typischerweise sehr klein angesetzt wird. In der Literatur wird in der Regel ein Takeover-Level von 1 angenommen, der aber nur durch Rundung erreicht wird.

Die rekursive Berechnung von $\vec{r}^{(t)} = (r_0^{(t)}, \ldots, r_{\lambda-1}^{(t)})$ kann simultan erfolgen:

$$\vec{r}^{(t)} = \overline{E}^{(t)\top} \, \vec{s}^{(t)} \, \overline{Q} \, . \tag{6.31}$$

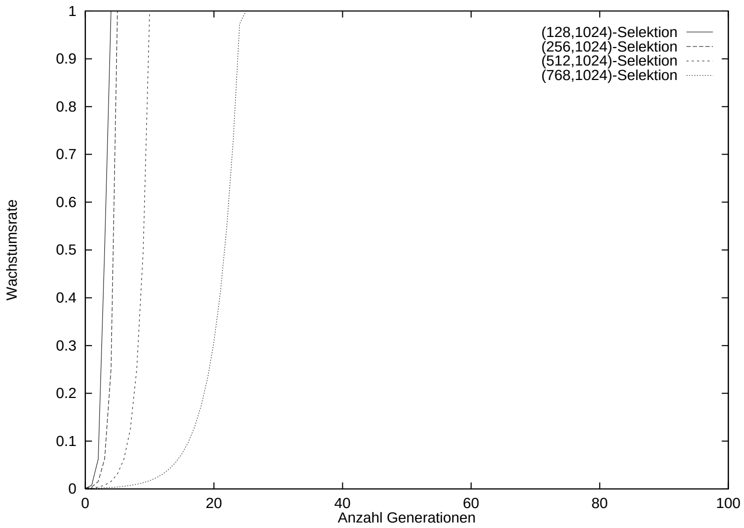

Abbildung 6.4: Wachstumskurven von panmiktischer (μ, λ)-Selektion.

Der Ausdruck $\overline{E}^{(t)\top}\ \vec{s}^{(t)} \in \mathbb{R}^{|\mathcal{E}|-1}$ liefert bereits die gewünschten Werte, allerdings bezogen auf die Deme. Die Multiplikation von rechts mit $\overline{Q}^{(t)}$ ordnet diese Werte schließlich wieder den einzelnen Individuen zu.

Der Wert $min\{s_i^{(t)}|i \in X\}$ kann als Wachstumskoeffizient betrachtet und, analog zu den in der Literatur üblichen Growth-Curves, als Wachstumskurve aufgetragen werden. Abb. 6.4 zeigt die gemäß Gl. 6.27 – 6.29 berechnete Wachstumskurve für eine panmiktische Population mit 1024 Individuen und (μ, λ)-Selektion. Als Erfolgsfunktion wurde wie bei Chakraborty, Deb und Chakraborty (1996)

$$p_{select}(\lambda, k) \quad = \quad \begin{cases} \frac{k}{\mu} & : \quad 0 \leq k \leq \mu \\ 1 & : \quad \text{sonst} \end{cases} \tag{6.32}$$

verwendet. Hierbei wird μ also als Parameter des Selektionsoperators aufgefaßt. Wie bei Goldberg und Deb (1991) scheint das Wachstum annähernd logistisch zu sein. Abb. 6.5 zeigt das Wachstum einer Population gleicher Größe mit Linear-Ranking-Selektion. Die Erfolgsfunktion ist wiederum dieselbe wie bei Chakraborty, Deb und Chakraborty (1996) und lautet

$$p_{select}(\lambda, k) \quad = \quad \frac{1}{\lambda}\left(\alpha - \frac{(\alpha-1)(k-1)}{\lambda - 1}\right) . \tag{6.33}$$

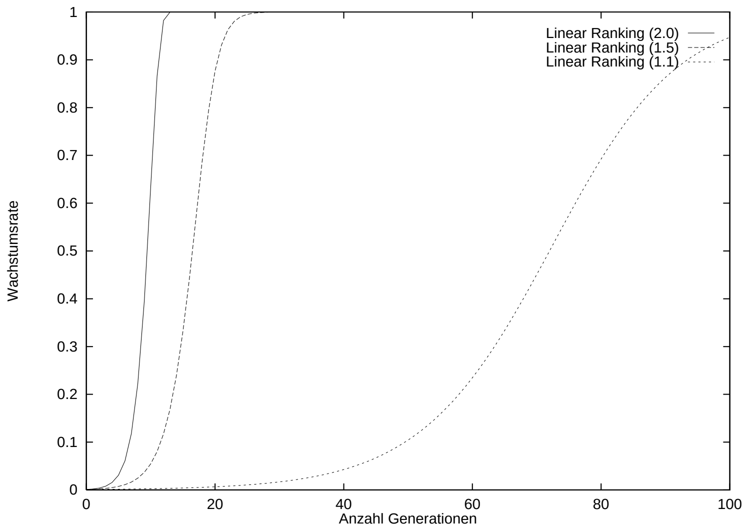

Abbildung 6.5: Wachstumskurven von panmiktischer linearer Rangselektion.

Die Wachstumskurven von Migrationsmodellen mit lokaler Kommaselektion (Abb. 6.6) und lokaler linearer Rangselektion (Abb. 6.7) zeigen die unterschiedliche Reaktion dieser beiden Selektionsverfahren bei verschiedenen Isolationszeiten. Die Rangselektion behält den logistischen Charakter auch bei längeren Isolationszeiten aufrecht, da diese entsprechend mehr Generationen zum lokalen Takeover benötigt. Der vergleichsweise starke Selektionsdruck einer Kommastrategie selbst mit relativ großer Elternzahl bewirkt, daß die Teilpopulationen schon bei kurzen Isolationszeiten in die Sättigung gehen. Das Nachbarschaftsmodell mit Rangselektion zeigt mit Ausnahme einer kurzen Anfangsphase und einer ebenfalls kurzen Sättigungsphase lineares Wachstum.

Der probabilistische Durchmesser stellt eine Erweiterung der Takeover-Time dar, und er teilt auch die Schwächen dieser Analysemethode. In den Abb. 6.4 bis 6.8 findet man mehrere Beispiel dafür, daß sehr verschiedene Wachstumskurven zu ähnlichen probabilistischen Durchmessern führen können. Daher ist letztlich die Wachstumskurve eher geeignet, einen Selektionsoperator zu charakterisieren.

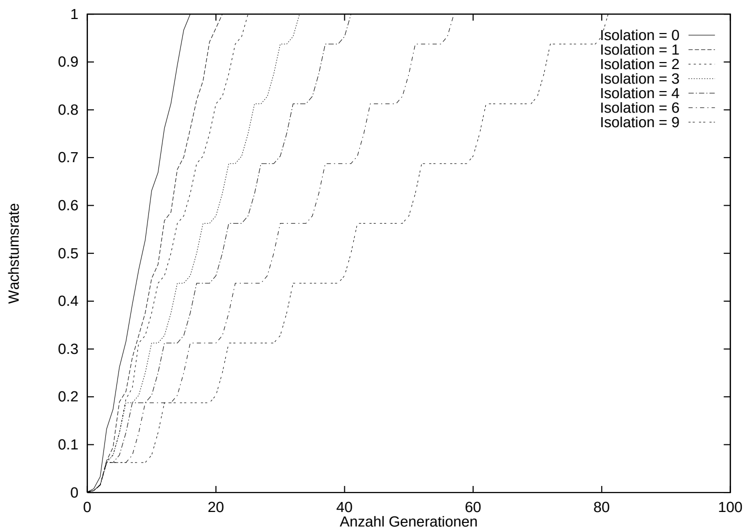

Abbildung 6.6: Wachstumskurven eines Migrationsmodells mit ringförmig angeordneten Teilpopulationen und lokaler $(16, 64)$-Selektion.

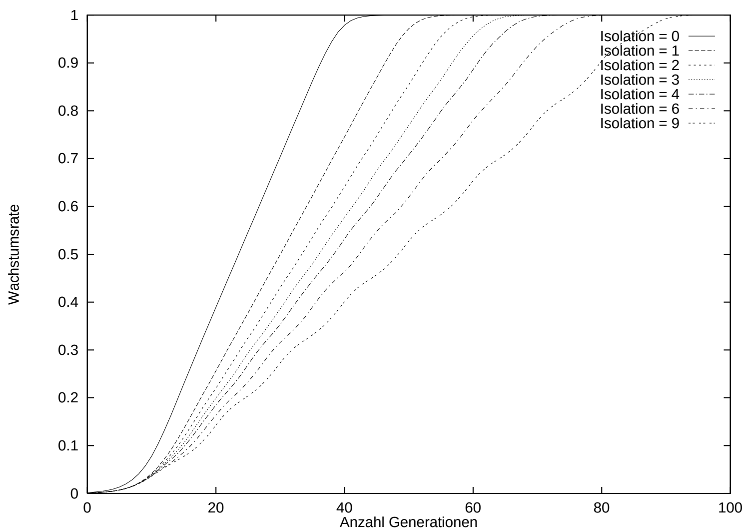

Abbildung 6.7: Wachstumskurven eines Migrationsmodells mit ringförmig angeordneten Teilpopulationen und linearer Rangselektion.

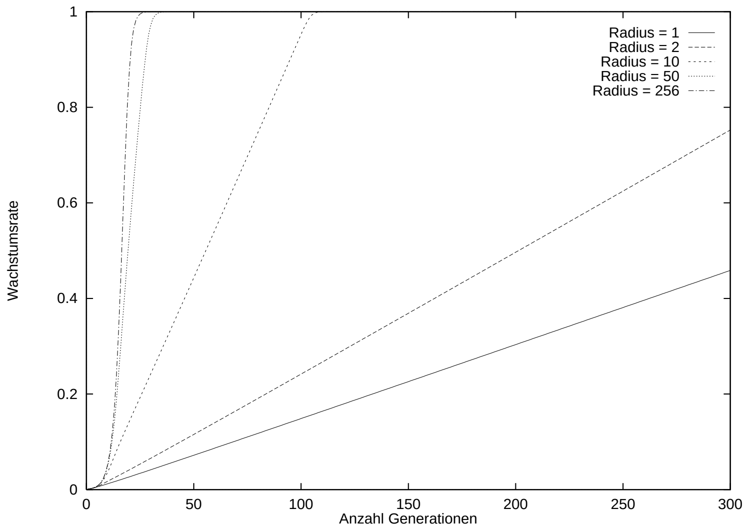

Abbildung 6.8: Wachstumskurven eines Nachbarschaftsmodells im Ring mit 1024 Individuen und linearer Rangselektion $(\alpha = 1.5)$.

6.3 Derandomisierung von Turnierselektion

Obwohl Turnierselektion eigentlich ein Selektionsoperator für panmiktische EA ist, besteht auf den zweiten Blick doch eine interessante Ähnlichkeit zu strukturierten, nichtpanmiktischen Populationen. Gordon (1994) hat einen panmiktischen GA parallelisiert, indem er die Population in gleichgroße Teilpopulationen aufteilt, auf diesen aber globale Turnierselektion durchführt. Immer, wenn ein Individuum gezogen wird, das nicht in der lokalen Population ist, findet Kommunikation statt. Wir wollen an dieser Stelle einen Schritt weiter gehen und Turnierselektion als nichtzyklische, dynamische Populationsstruktur definieren. Gegeben sei eine Population $P = (p_0^{(t)}, \ldots, p_{\lambda-1}^{(t)})$ zum Zeitpunkt t. Zur Turnierselektion werden die potentiellen Eltern des Individuums $p_i^{(t+1)}$ durch gleichverteiltes Ziehen von k Eltern $\{r_{i,0}, \ldots, r_{i,k-1}\}$ aus $P^{(t)}$ ermittelt, wobei k die Turniergröße ist. Man erhält also eine temporäre Populationsstruktur der folgenden Art:

$$\begin{aligned}
\Pi_{\text{TS}} &= (X, \mathcal{E}^{(t)}, \mathcal{Q}) \\
X &= \{0, \ldots, \lambda - 1\} \\
\mathcal{Q} &= \Big\{\{0\}, \ldots, \{\lambda - 1\}\Big\} \\
\mathcal{E} &= (E_0, \ldots, E_{\lambda-1}) \\
E_i^{(t)} &= (r_{i,0}^{(t)}, \ldots, r_{i,k-1}^{(t)}), r_{i,j}^{(t)} \in \lambda\mathbb{Z}
\end{aligned} \tag{6.34}$$

Man kann zusätzlich fordern, daß die $r_{i,j}^{(t)}$ paarweise verschieden sind und erhält so Turnierselektion ohne Zurücklegen. Verwendet man eine einmal generierte, feste Folge von Zufallszahlen für die $r_{i,j}^{(t)}$, so hat man den Zufall aus dem Selektionsprozeß herausgezogen.

Man kann nun überlegen, ob man nicht auch mittels einer zyklischen Populationsstruktur eine hinreichende Annäherung der Turnierselektion erzielen kann. Dabei wollen wir zunächst sogar von einer statischen Populationsstruktur ausgehen. Natürlich kann man hierzu nicht einfach einen zufälligen Hypergraphen mit $|E_i| = k$ generieren, da dieser nicht notwendigerweise zusammenhängend ist. Um die relativ schnelle Ausbreitung guter Individuen im Falle zufälliger Turniere zu ermöglichen, sollte daher ein Hypergraph mit geringem Durchmesser gewählt werden.

Da für jedes zu ersetzende Individuum ein anderes Turnier durchgeführt werden soll, besteht die Partition Q aus den Mengen $\{i\}, i = 0, \ldots, \lambda - 1$. Gesucht ist also ein zusammenhängender Hypergraph mit $|E_i| = k$, möglichst kleinem Durchmesser, wobei idealerweise $m(H(x))$, also die Anzahl der Kanten, die x enthalten, für alle $x \in X$ gleich ist.

Ein naheliegender Vorschlag für einen solchen Hypergraphen ist ein Ring mit Abkürzungen. Sei $k \in \lambda\mathbb{Z}$ die Turniergröße. Dann besitzt der Hypergraph $H = (X, \mathcal{E}, \mathcal{Q})$ mit

$$\begin{aligned} X &= \{0, \ldots, \lambda - 1\} \\ \mathcal{Q} &= \Big\{\{0\}, \ldots, \{\lambda - 1\}\Big\} \\ \mathcal{E} &= (E_0, \ldots, E_{\lambda-1}) \\ E_i &= \{i + 1\} \cup \{i + \nu \cdot \tfrac{\lambda}{k-1} \,|\, \nu = 0, \ldots, k - 2\} \end{aligned} \tag{6.35}$$

den Durchmesser $\delta(H) = 1 + \lambda/(2(k - 1))$. Da wir es nun mit einer deterministischen Selektion in Verbindung mit einer Populationstruktur haben, liefert der Durchmesser der Struktur zugleich auch die Takeover-Time. Diese können wir nun mit einer Abschätzung der Takeover-Time für die Turnierselektion von Goldberg und Deb (1991) vergleichen. Abb. 6.9 zeigt, daß die statische Populationsstruktur aus Gl. 6.35 bei kleinen Turniergrößen eine Takeover-Zeit aufweist, die signifikant über der Abschätzung für die klassische Turnierselektion liegt. Man kann daraus schließen, daß die gewählte statische Populationsstruktur einen zu hohen Grad an Lokalität aufweist, um panmiktische Turnierselektion adäquat anzunähern. Eine bessere Approximation der zufallsgesteuerten Turnierselektion erreicht man durch eine dynamische Populationsstruktur. Verwendet man für jede Generation einen zufälligen Hypergraphen $\mathcal{E}^{(t)}$ mit $|E_i| = k, i \in \lambda\mathbb{Z}$, so erhält man genau eine Realisierung einer zufälligen Turnierselektion. Es reicht aber bereits aus, eine zyklische Populationsstruktur mit kurzer Periode zu verwenden, wie Abb. 6.10 zeigt. Mit der folgenden Populationsstruktur kann man eine deter-

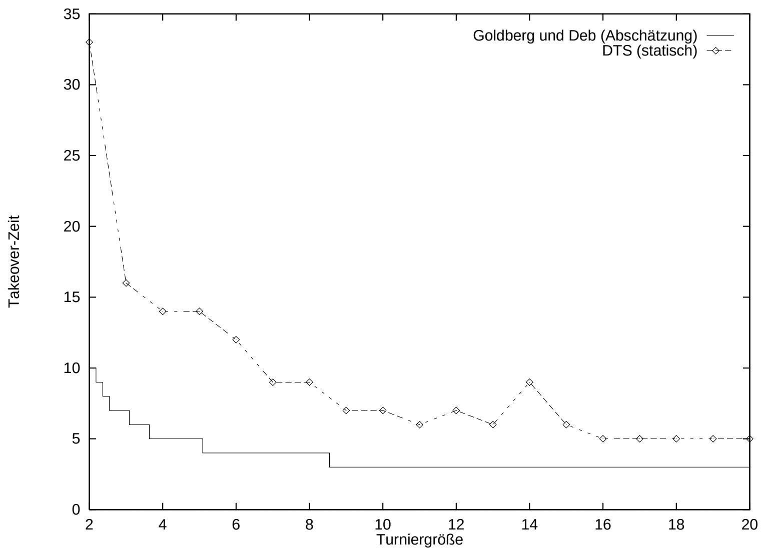

Abbildung 6.9: Takeover-Zeiten von Turnierselektion (Abschätzung) und derandomisierter Turnierselektion

ministische Turnierselektion (DTS) erreichen:

$$\begin{aligned}
\Pi_{\text{DTS}} &= (X, \mathcal{E}, \mathcal{Q}) \\
X &= \{0, \ldots, \lambda - 1\} \\
\mathcal{Q} &= \{\{0\}, \ldots, \{\lambda - 1\}\} \\
\mathcal{E}^{(t)} &= (E_0^{(t)}, \ldots, E_{\lambda-1}^{(t)}) \\
E_i^{(t)} &= \Big\{ i + \lceil ((jc + t)\ v) \rceil \bmod \lambda \,\Big|\, j = 1, ..., k-1) \Big\} \\
&\quad \cup \{i + 1 \bmod \lambda\},\ i = 0, ..., \lambda - 1 \\
v &= \tfrac{\lambda}{kc+1}\,,
\end{aligned} \qquad (6.36)$$

wobei c die Periode der Populationsstruktur ist. Dadurch, daß der direkte Nachbar $i + 1 \bmod \lambda$ in jedem Turnier enthalten ist, ist gewährleistet, daß die Populationsstruktur zusammenhängend ist.

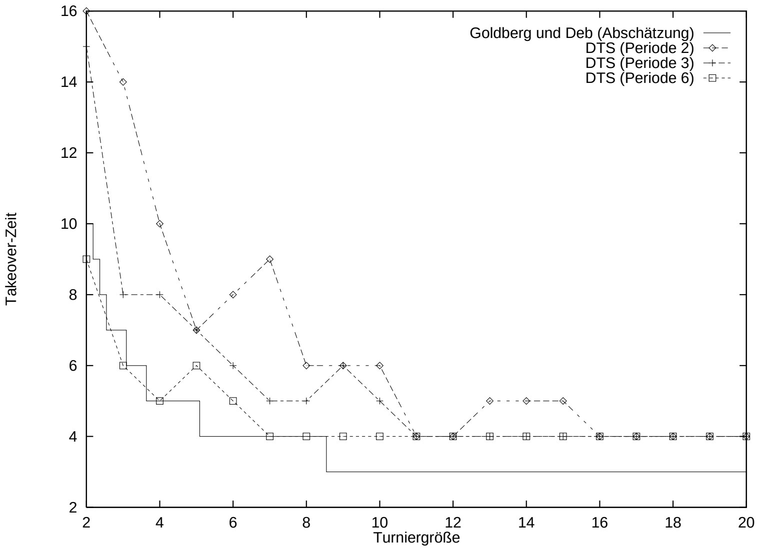

Abbildung 6.10: Takeover-Zeiten von zufälliger Turnierselektion (geschätzt) und deterministischer Turnierselektion mit Perioden $2, 3$ und 6.

Kapitel 7

Experimenteller Vergleich von Populationstrukturen

7.1 Ziele der Experimente

In Kap. 4 wird ein formales Modell für Populationsstrukturen definiert. Die zum Erreichen eines gewissen Abstraktionsgrades notwendige Generalisierung der Modellierung hat zur Folge, daß nicht alle in der Literatur veröffentlichten Mechanismen im Zusammenhang mit Populationsstrukturen abgebildet werden können. Insbesondere werden in der Modellierung Individuen innerhalb einer Teilpopulation als ununterscheidbar angesehen. Dies führt dazu, daß beim Austausch zwischen Teilpopulationen keine zusätzliche Selektion erfolgt, so daß beispielsweise bei einem Migrationsmodell die Emigranten immer zufällig ausgewählt werden. Ein wesentliches Ziel der Experimente war es daher, die Auswirkungen der Selektion von Migranten auf den Optimierungsverlauf in Abhängigkeit vom gewählten Populationsmodell zu untersuchen.

Ein weiterer Aspekt der Experimente war die Überprüfung der verbreiteten These, daß nichtpanmiktische Modelle besonders für multimodale Zielfunktionen geeignet sind, während sie im unimodalen Fall qualitativ unterlegen sind.

In den unten dokumentierten Parameterstudien wird für drei Zielfunktionen untersucht, wie vier häufig verwendete Populationsstrukturen auf die Veränderung der Populationsgröße reagieren. Die verwendeten Populationsstrukturen sind Panmixie (Gl. 4.16) sowie Migration (Gl. 4.17) und Pollination (Gl. 4.18) mit r Teilpopulationen mit je ν Individuen und Migrationsmengen

$$M_{s\to t} = \begin{cases} s\nu & : \; s \equiv t-1 \mod r \\ s\nu +_\lambda 1 & : \; s \equiv t+1 \mod r \\ \emptyset & : \; \text{sonst.} \end{cases} \tag{7.1}$$

$$s,t \in r\mathbb{Z} \tag{7.2}$$

sowie ein Nachbarschaftsmodell im Ring nach Sprave (1994) mit Nachbarschaftsradius $\rho = 1$. Bei den Multipopulationsmodellen wurde die Populationsgröße mittels der Anzahl der Teilpopulationen variiert. Um zu erkennen, ob Veränderungen allein durch die Populationsgröße oder auch durch die Struktur bewirkt wurden, wurde zum Vergleich jeweils das panmiktische Modell in Form einer $(r\cdot\mu, r\cdot\lambda)$-ES ausgewertet, wobei r der Anzahl der Teilpopulationen in den Multipopulationsmodellen entspricht. Beim Nachbarschaftsmodell im Ring wurde der Nachkommenüberschuß von λ/μ explizit erzeugt, daher wurde ein Ring mit $r\cdot\mu$ Individuen verwendet.

Für die beiden Multipopulationsmodelle wurde schließlich noch unterschieden, in welcher Weise die Individuen ausgewählt werden, die in eine Nachbarpopulation verschoben oder kopiert werden. Bei einer Implementierung gemäß der Modellierung in Kap. 4 werden Individuen zu Migranten, indem sie per Geburt auf bestimmte Indizes gesetzt werden. Da die Reihenfolge, in der Individuen aus einer gemeinsamen Deme erzeugt werden, das Resultat eines Zufallsprozesses ist, kann durch eine explizite Implementierung der Populationsmodelle (s. Kap. 4.5) auch nur eine zufällige Auswahl der Migranten erreicht werden. Austauschstrategien, die in irgendeiner Weise die Migranten anhand ihrer Qualität auswählen, können hingegen nicht modelliert werden. Die verbreitete Alternative, das beste Individuum einer Teilpopulation zu versenden, wurde daher zusätzlich untersucht.

Bei allen Experimenten wurden bis auf die Populationsstruktur klassische ES-Operatoren gemäß Kap. 2.3.3 verwendet. Jedes Individuum verfügte über genau eine Schrittweite als Strategieparameter. Es wurde Rekombination aus zwei Eltern verwendet, wobei die Objektvariablen diskret und die Schrittweiten intermediär (arithmetisch) rekombiniert wurden.

7.2 Populationsstrukturen und das Kugelmodell

Das Kugelmodell, formal

$$f_{sphere}(\vec{x}) = \sum_{i=1}^{n} x_i^2 \rightarrow min! \quad (7.3)$$
$$\vec{x} \in \mathbb{R}^n , \quad (7.4)$$

kann aus zwei Gründen als Referenzproblem für die Optimierung auf konvexen Mengen angesehen werden. Zum einen ist es die mit Abstand am besten untersuchte kontinuierliche Zielfunktion im Bereich der Theorie der Evolutionären Algorithmen (Schwefel 1977; Beyer 1997). Zum anderen existiert die These, daß viele kontinuierliche Optimierprobleme in hinreichender Nähe eines Optimums annähernd quadratischer Natur sind[1]. Daher gibt das Verhalten eines direkten Optimierverfahrens auf dem Kugelmodell aufschluß über dessen Eignung als lokales Suchverfahren.

Im Hinblick auf die Parallelisierung Evolutionärer Algorithmen wird untersucht, wie die Fortschrittsgeschwindigkeit einer panmiktischen und mehrerer nichtpanmiktischer Evolutionsstrategien mit der gleichzeitigen Erhöhung der Eltern- und Nachkommenzahl skaliert.

Von besonderem Interesse ist dabei die Frage, ob strukturierte Populationsmodelle ein qualitativ ähnliches Verhalten wie der panmiktische Grundalgorithmus aufweisen. In Abbildung 7.1 ist für jedes der getesteten Populationsmodelle die, über 250 Versuche gemittelte, Anzahl der Zielfunktionsauswertungen aufgetragen, die benötigt wurden, um vom Startpunkt $\vec{x} = (1, \ldots, 1)^\top$ zum ersten Mal einen Punkt

[1] Auf dieser Modellannahme basiert z.B. das Newton-Raphson-Verfahren.

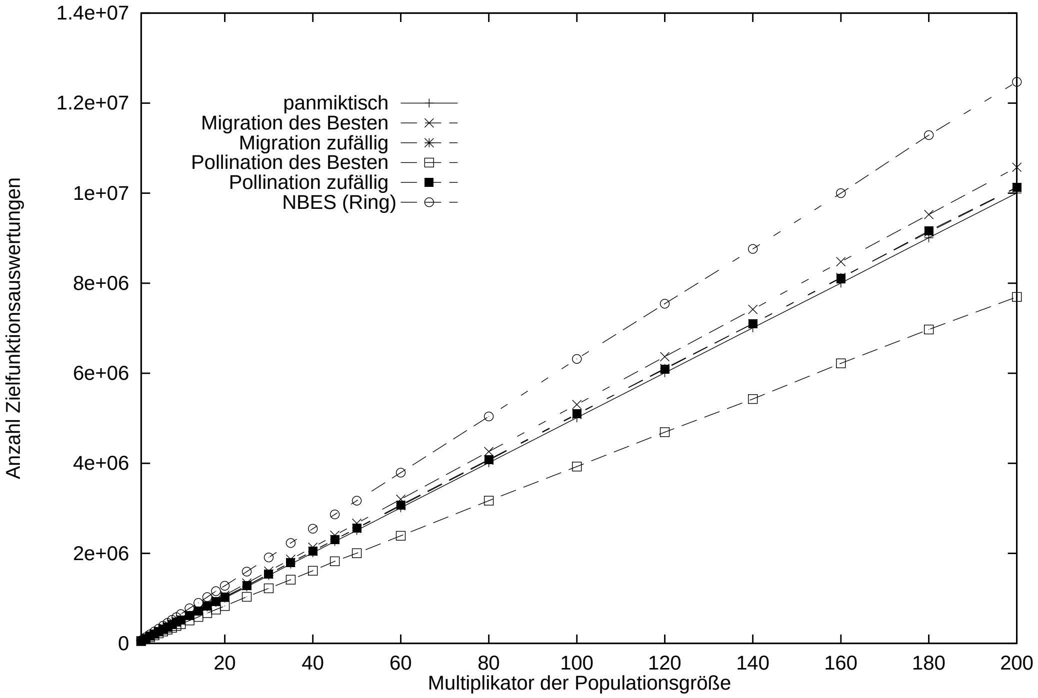

Abbildung 7.1: Populationsmodelle auf dem Kugelmodell. Aufgetragen ist die Anzahl der für die Erreichung von $f(\vec{x}) < 10^{-20}$ benötigten Zielfunktionsauswertungen gegen die Anzahl der Teilpopulationen (Migration, Pollination) bzw. Vervielfachung der Population (Panmixie, Nachbarschaftsmodell).

$\vec{x}$ mit $f_1(\vec{x}) < 10^{-20}$ zu finden. Die Experimente wurden durchgeführt mit

$$
\begin{aligned}
n &= 30 \\
\mu &= 15 \\
\lambda &= 105 \\
r &= 1, 2, \ldots, 10, \\
&\quad\; 12, 14, \ldots, 20, \\
&\quad\; 40, 60, \ldots, 200 \\
\tau_0 &= 0,1\ .
\end{aligned}
\tag{7.5}
$$

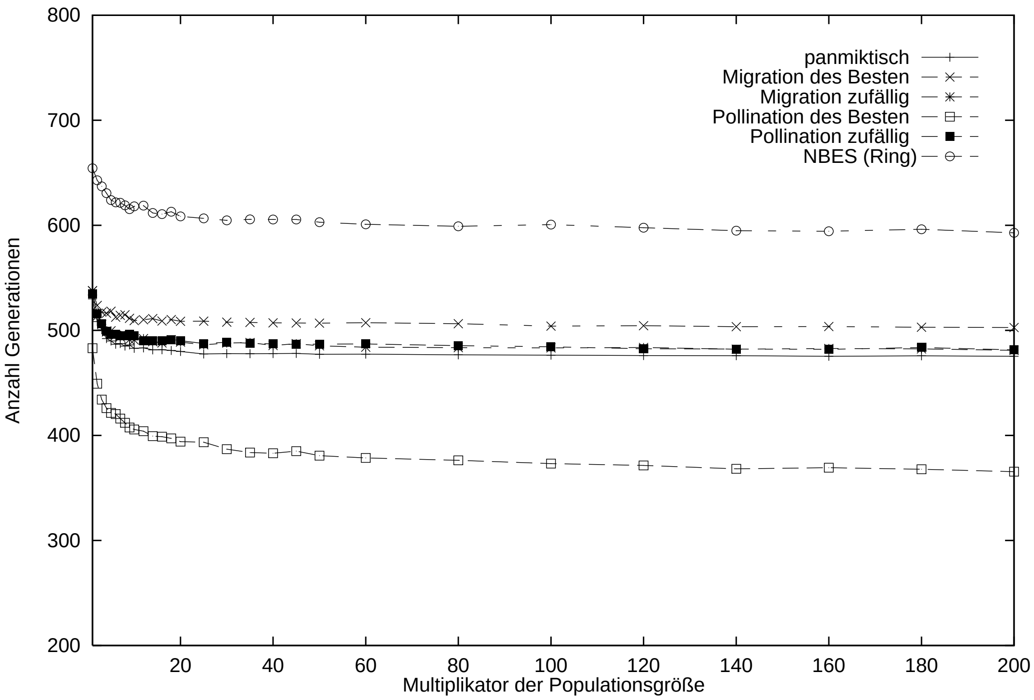

Abbildung 7.2: Die Ergebnisse aus Abb. 7.1, jedoch aufgetragen gegen die benötigte Anzahl von Generationen.

Die beiden Multipopulationsmodelle mit zufälligem Austausch verhalten sich nahezu identisch und benötigen geringfügig mehr Auswertungen als der Standardalgorithmus zur Erreichung des vorgegebenen Ziels. Migration des besten Individuums schneidet noch etwas schlechter ab, liegt aber ebenfalls sehr nahe am panmiktischen Fall.

Das Nachbarschaftsmodell benötigt mehr als ein Drittel mehr Auswertungen, zeigt aber qualitativ dasselbe Verhalten wie eine panmiktische ES. Die geringe Anzahl von Zielfunktionsauswertungen, die das Pollinationsmodell mit Austausch des besten Individuums benötigt, ist nur auf den ersten Blick erstaunlich. Man muß berücksichtigen, daß bei jedem Pollinationsvorgang das beste Individuum jeder Teilpopulation zweifach kopiert wird. Zusätzlich bleibt das Original mit Wahrscheinlichkeit $(1 - 1/\lambda)^2$ erhalten. Damit wird den besten Individuen der Teilpopulationen ein starker Selektionsvorteil eingeräumt, der von der gleichverteilten Selektion der klassischen ES stark abweicht. Bei echter Migration hingegen bleiben alle Individuen in gleicher Zahl erhalten, so daß, entgegen der Intuition, auch diese Austauschstrategie als ungerichtet angesehen werden kann.

Besser als in Abb. 7.1 sieht man in Abb. 7.2, daß die Anzahl der benötigten Zielfunktionsauswertungen nicht linear mit der Populationsgröße steigt. Bei allen Populationsmodellen sinkt die Anzahl der benötigten Generationen mit größer werdender Population.

Wir gelangen daher zu der folgenden These:

Die lokalen Sucheigenschaften der klassischen Evolutionsstrategie bleiben beim Übergang auf in sich panmiktische Teilpopulationen mit zufälligem Austausch qualitativ erhalten.

Mit anderen Worten, die Verwendung von nichtpanmiktischen ES schadet nicht, auch wenn das Problem relativ einfach ist. Vielmehr kann man davon ausgehen, daß in der Nähe eines Optimums die Fähigkeit der ES zum schnellen Abstieg erhalten bleibt.

7.3 Multimodale Zielfunktionen

Die folgenden Versuche sollen einen ersten Einblick geben, welchen Einfluß die Auswahl der Migranten auf die Ergebnisqualität bei der Optimierung multimodaler Zielfunktionen hat. Die Experimente wurden wiederum mit ES-Varianten auf zwei Zielfunktionen durchgeführt, der verallgemeinerten Rastrigin-Funktion (Gl. 7.7) und der Keaneschen Bump-Funktion (Gl. 7.8). Gemessen wurde die relative Häufigkeit, mit der das globale Optimum bzw. der beste bekannte Zielfunktionswert gefunden wurde, wenn ein Verfahren 500 mal mit unterschiedlicher Initialisierung des Pseudozufallszahlengenerators gestartet wurde. Zur Variation der

Populationsgröße wurden folgende Parameter verwendet:

$$\begin{aligned} n &= 30 \\ \mu &= 15 \\ \lambda &= 105 \\ r &= 1, 2, \ldots, 10, \\ &\quad 12, 14, \ldots, 20, \\ &\quad 25, 30, \ldots, 50, \\ &\quad 60, 80, \ldots, 200 \\ &\quad 400, 600, \ldots, 1000\,. \end{aligned} \tag{7.6}$$

Eine der meistverwendeten Zielfunktionen zur Analyse Evolutionärer Algorithmen ist die von Rudolph (1990) modifizierte Rastrigin-Funktion (Rastrigin 1974). Diese Funktion stellt eine Überlagerung aus einem Kugelmodell und einer Kosinusschwingung dar. Die Amplitude der Schwingung ermöglicht es, die Welligkeit der Zielfunktionstopologie einzustellen. Rudolph (1990) verwendete diese Funktion, um zu zeigen, daß eine nach dem Migrationsmodell parallelisierte Evolutionsstrategie in der Lage ist, mit hoher Sicherheit das globale Optimum einer multimodalen Zielfunktion zu finden. Sprave (1990) konnte dies auch für einen Genetischen Algorithmus mit einem Nachbarschaftsmodell zeigen. Die Rastrigin-Funktion

$$\begin{aligned} f_{rast}(\vec{x}) &= nA + \textstyle\sum_{i=1}^{n} \left[x_i^2 - A\cos(\omega x_i)\right] \;\rightarrow min\,! \\ &\quad \vec{x} \in \mathbb{R}^n \end{aligned} \tag{7.7}$$

wurde für diese Studie mit der Parametrisierung $n = 30, A = 10$ und $\omega = 2\pi$ verwendet. Das Minimum dieser Funktion von $f_{rast}(\vec{x}^*) = 0$ wird bei $\vec{x}^* = (0, \ldots, 0)^\top$ angenommen.

Die Initialisierung der Individuen erfolgte von dem festen Startpunkt $\vec{x}_0 = (-10, \ldots, -10)^\top$ aus. Zur Ermittlung der Erfolgsrate wurde das Optimum bei einem Zielfunktionswert 0.01 oder kleiner als gefunden angesehen.

Abb. 7.3 zeigt die Erfolgsrate der getesteten Populationsstrukturen bei einer vorgegebenen Anzahl von Zielfunktionsauswertungen von $2.000.000$ Aufrufen, aufgetragen gegen den Multiplikator der Populationsgröße. Die Begrenzung der Zielfunktionsaufrufe bewirkt, daß ab einer gewissen Populationsgröße die Anzahl der Generationen nicht mehr ausreicht, um das globale Optimum zu erreichen. Daher sinkt die Erfolgsrate für alle Verfahren oberhalb einer Populationsgröße von ca. $\lambda = 10500$ $(r = 100)$ auf Null. Bemerkenswert ist, daß keines der nichtpan-

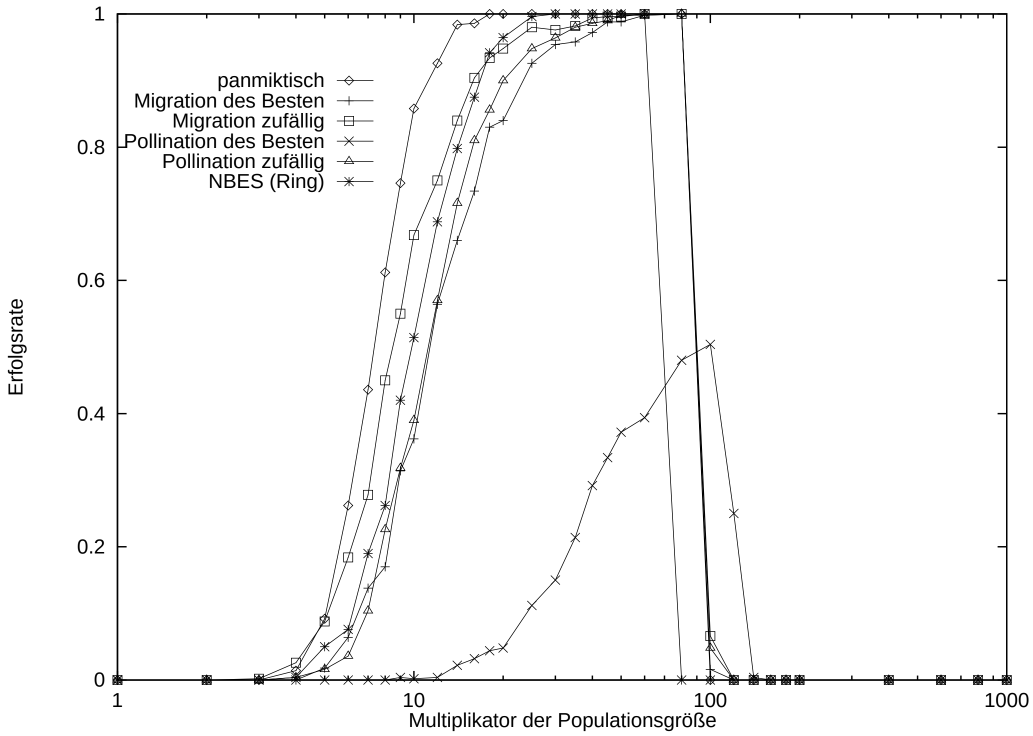

Abbildung 7.3: Erfolgsrate verschiedener Populationsstrukturen auf der verallgemeinerten Rastrigin-Funktion. Bei fester Anzahl von Zielfunktionsauswertungen wurde die Anzahl der Teilpopulationen der Multipopulationsmodelle variiert. Beim panmiktischen Modell und beim Nachbarschaftsmodell wurde die Populationsgröße entsprechend angepaßt.

miktischen Populationsmodelle bei gleicher Populationsgröße die Erfolgsrate des panmiktischen Modells erreicht (außer bei einer Erfolgsrate von 1.0). Vor dem

Hintergrund, daß diese Funktion oft als Beispiel für eine qualitative Verbesserung von EA durch Parallelisierung verwendet wird, kann man schließen, daß der Gewinn überwiegend durch die Populationsgröße und nicht durch die Strukturierung erzielt wird. Migration und Pollination mit zufälliger Auswahl der Migranten sowie Migration des besten Individuums liegen sehr nahe beieinander, und zwar knapp unter dem panmiktischen Modell. Das Verhalten des Nachbarschaftsmodells ist bei dieser Zielfunktion nahezu identisch mit dem der Multipopulationsmodelle.

Deutliche Abweichungen hingegen sieht man in Abb. 7.3 bei der Pollination des Besten. Ähnlich wie beim Kugelmodell bewirkt die starke Bevorzugung der lokal besten Individuen einen schnellen Abstieg in lokale Optima, in denen das Verfahren dann aufgrund des Diversitätsverlustes stagniert. Andererseits erzielt es bis zum Faktor 200 noch eine Erfolgsrate von ca. 0.25. Dies folgt daraus, daß Pollination des Besten im Erfolgsfall durchschnittlich weniger Generationen zum Auffinden des Optimums benötigte als die anderen Kombinationen aus Populationsmodell und Austauschstrategie.

Die Unterschiede zwischen den Multipopulationsmodellen verschwinden fast völlig, wenn Isolationsphasen eingeführt werden. In Abb. 7.4 (20 Generationen Isolationszeit) und 7.5 (50 Generationen Isolationszeit) sieht man einerseits ein beinahe unverändertes Verhalten von zufälliger Migration und Pollination sowie der Migration des Besten. Die Strategie, die mit Pollination des Besten arbeitet, rückt mit steigender Isolationszeit näher an die anderen Verfahren heran. Etwa im gleichen Maße schwindet der Geschwindigkeitsvorteil.

Die zweite Zielfunktion dieser Studie wurde von Keane (1994) aufgestellt, um Optimierverfahren im Hinblick auf ihre Eignung zur Optimierung bestimmter Klassen von strukturellen Designaufgaben zu testen. Da viele der von ihm bearbeiteten Anwendungen multimodal und restringiert waren, stellte er folgende Testfunktion auf, die als *Keane's Bump Function* mittlerweile in einigen Untersuchungen zu finden ist, wie z.B. (Michalewicz, Nazhiyath und Michalewicz 1996):

$$\begin{aligned} f_{keane}(\vec{x}) &= -\left|\frac{\sum_{i=1}^{n} cos^4(x_i) - 2\prod_{i=1}^{n} cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} i x_i^2}}\right| \rightarrow min\,! \\ & \vec{x} \in \mathbb{R}^n \end{aligned} \tag{7.8}$$

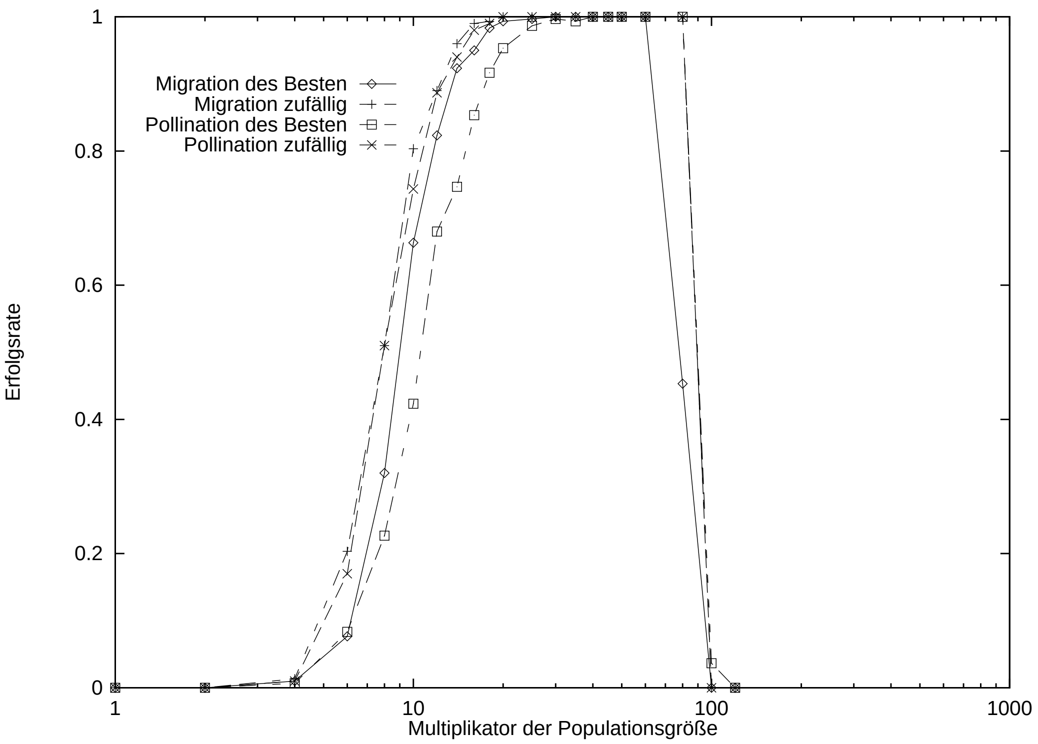

Abbildung 7.4: Erfolgsrate der getesteten Multipopulationsmodelle auf der Rastrigin-Funktion mit einer Isolationszeit von 20 Generationen.

mit den Restriktionen

$$\prod_{i=1}^{n} x_i \geq 0.75 \tag{7.9}$$

$$\sum_{i=1}^{n} x_i \leq 7.5n \tag{7.10}$$

$$0 \leq x_i \leq 10,\ i = 1, \ldots, n\,. \tag{7.11}$$

Diese Funktion weist eine wellige Anwortoberfläche auf, deren Täler[2] zum Ursprung hin tiefer werden. Daher liegen die tiefsten Stellen nahe an der hyperbo-

[2]Ursprünglich war diese Funktion als Maximierungsaufgabe formuliert. Die Vertiefungen sind also eigentlich Buckel (bumps).

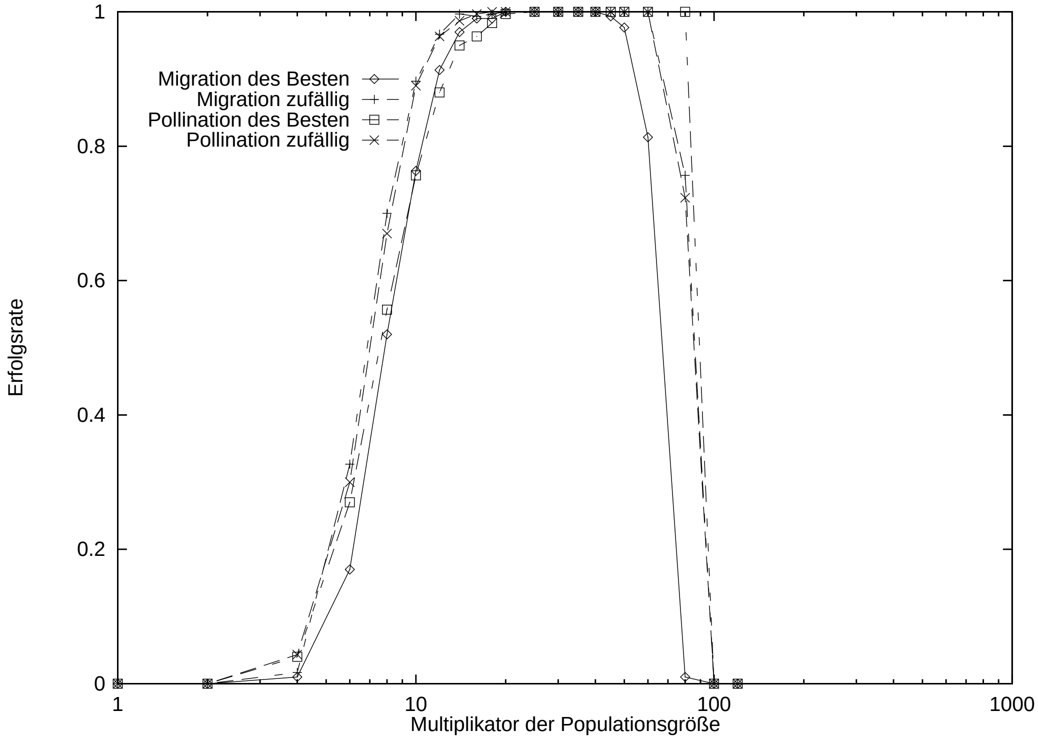

Abbildung 7.5: Erfolgsrate der getesteten Multipopulationsmodelle auf der Rastrigin-Funktion mit einer Isolationszeit von 50 Generationen.

loiden Restriktion aus Gl. 7.9. Das globale Optimum ist nicht bekannt, der beste bekannte Zielfunktionswert für die in diesen Experimenten verwendete Dimension von $n = 20$ liegt ungefähr bei -0.803. Dieser Wert wurde bisher nur an einer Stelle im Suchraum gefunden (einschließlich der hier dokumentierten Experimente). Gemessen wurde die relative Häufigkeit, mit der ein Funktionswert von -0.8 unterschritten wurde. Die Restriktionen wurden gemäß Abschnitt 8.1 mit der metrischen Straffunktion aus Gl. 8.17 behandelt. Die Initialisierung der x_i erfolgte gleichverteilt aus dem Intervall$[0, 10]$.

Im Gegensatz zur modifizierten Rastrigin-Funktion ist bei der Keaneschen Bump-Funktion die Populationsgröße nicht der entscheidende Parameter (Abb. 7.6). Die panmiktische ES kommt über eine Erfolgsrate von 0.25 nicht hinaus. Alle Multipopulationsverfahren erreichen mit einer großen Anzahl von Populationen nahezu

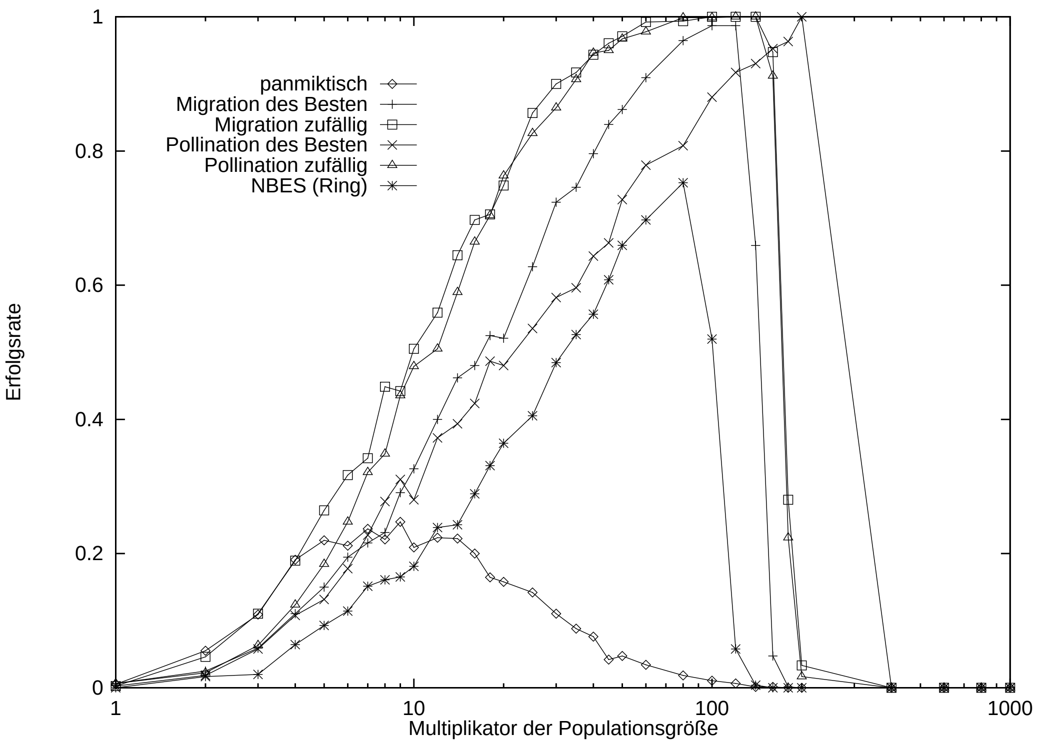

Abbildung 7.6: Erfolgsrate verschiedener Populationsstrukturen auf der Keaneschen Bump-Funktion. Bei fester Anzahl von Zielfunktionsauswertungen wurde die Anzahl der Teilpopulationen der Multipopulationsmodelle variiert. Beim panmiktischen Modell und beim Nachbarschaftsmodell wurde die Populationsgröße entsprechend angepaßt.

eine Erfolgsrate von 1. Die Verfahren mit der Auswahl des Besten weisen jedoch durchweg schlechtere Erfolgsraten auf.

In Verbindung mit Isolationsphasen (Abb. 7.7 und 7.8) erzielen alle Multipopulationsverfahren geringere Erfolgsraten als ohne Isolation. Wiederum ist kein großer Unterschied zwischen den Verfahren mit zufälligem Austausch zu beobachten. Die Pollination des Besten verschlechtert sich in geringerem Maße als die anderen Verfahren, während die Migration des Besten sich mit zunehmender Isolationsdauer weiter verschlechtert.

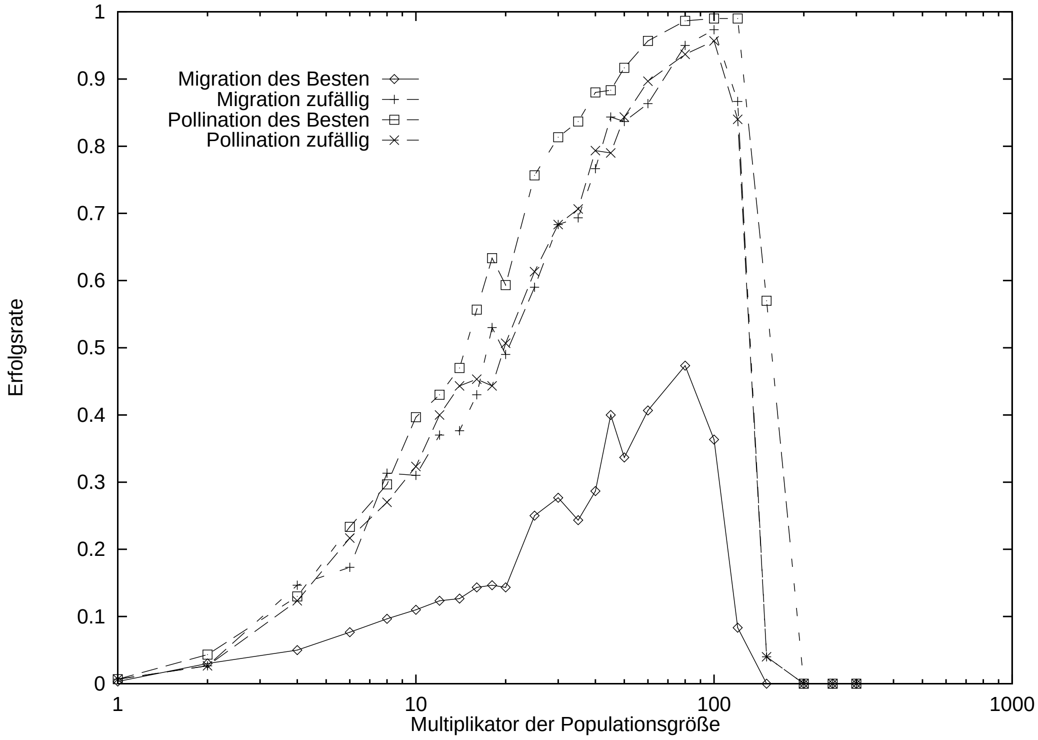

Abbildung 7.7: Erfolgsrate der getesteten Multipopulationsmodelle auf der Keaneschen Bump-Funktion mit einer Isolationszeit von 20 Generationen.

7.4 Bewertung der Experimente

Evolutionäre Algorithmen sind aufgrund ihres Populationskonzepts eine Kombination aus volumenorientierter Suche (Exploration des Suchraums) und pfadorientierter Suche (Ausbeutung lokaler Optima). Die hier diskutierten Versuche geben Hinweise darauf, daß man explorativen Charakter von Evolutionsstrategien verstärken kann, ohne die Fähigkeit zur lokalen Optimierung zu verlieren. Aus der Sicht der Parallelisierung bedeutet dies, daß Multipopulationsmodelle eine gleich schnelle, effizient parallelisierbare Alternative zur panmiktischen ES darstellen[3].

[3]Der Begriff des linearen Speed-Up wird hier bewußt vermieden, da es sich um Algorithmen mit unterschiedlichem Ein-Ausgabeverhalten handelt.

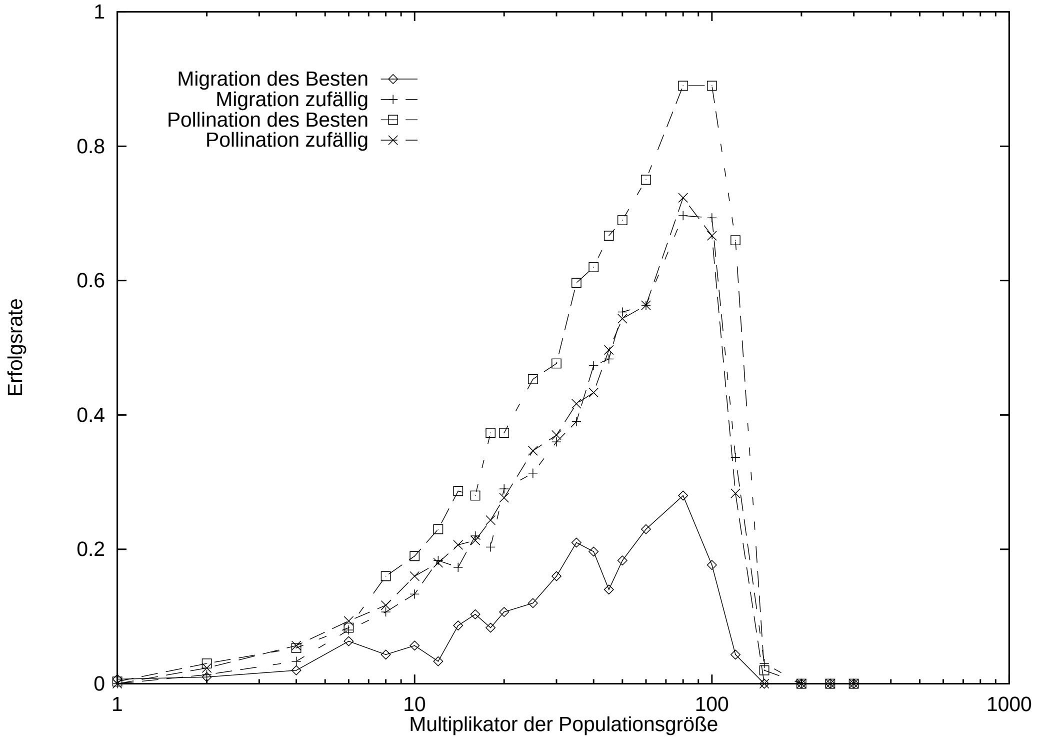

Abbildung 7.8: Erfolgsrate der getesteten Multipopulationsmodelle auf der Keaneschen Bump-Funktion mit einer Isolationszeit von 50 Generationen.

Bei den multimodalen Zielfunktionen befinden sich die Individuen verschiedener Teilpopulationen eines Multipopulationsmodells häufig in den Attraktionsgebieten verschiedener lokaler Optima. Der Austausch von Individuen führt dann bei Verwendung von Rekombination zur Rekombination auf lokalen Optima. Dabei hängt es von der Zielfunktion und vom Rekombinationsoperator ab, ob die Rekombination zweier lokaler Optima ein anderes, möglicherweise besseres ergibt, oder ob sie, bildlich gesprochen, auf den Grat zwischen zwei Tälern führt. Die an den Koordinatenachsen ausgerichtete, gitterförmige Anordnung der lokalen Optima bei der Rastrigin-Funktion steht bei diskreter Rekombination für den ersten Fall. Bei der Keaneschen Funktion ist nicht nur diese Regelmäßigkeit nicht gegeben, vielmehr birgt die Nähe des (mutmaßlichen) globalen Optimums zum unzulässigen Gebiet die Gefahr, daß durch Rekombination Restriktionen verletzt werden. Hierin liegt eine mögliche Erklärung für die Beobachtung, daß eine ES auf der Rastrigin-Funktion von Isolation profitiert, auf der Keaneschen Funktion hingegen schlechter wird: Bei letzterer kann durch häufigen Austausch verhindert werden, daß die Teilpopulationen sich so weit voneinander entfernen, daß der Austausch von Individuen mehr schadet als nutzt. Bei der Rastrigin-Funktion hingegen funktioniert die Rekombination auf lokalen Optima umso besser, je genauer diese angenähert sind.

Diese Studie zeigt, daß die Auswahl der zu versendenden Individuen großen Einfluß auf die Ergebnisqualität eines Multipopulationsmodells haben kann. Man könnte nun daraus folgern, daß man ein Modell für Populationsstrukturen benötigt, das diese Effekte berücksichtigt. Andererseits führt die nichtzufällige Auswahl der Migranten zu einer Überlagerung mit dem eigentlichen Selektionsoperator, so daß nicht mehr unterschieden werden kann, ob das veränderte Verhalten tatsächlich durch die Austauschstrategie bewirkt wird, oder ob es überwiegend auf die resultierende Selektionsverteilung zurückzuführen ist. Da aber strukturierte Populationen meist gerade zur Vermeidung vorzeitiger Stagnation in einem nichtglobalen Optimum eingesetzt werden, ist eine zusätzliche Bevorzugung guter Individuen kontraintuitiv.

Kapitel 8

Weitere Konzepte

8.1 Restriktionsbehandlung für rangbasierte Algorithmen

In der Anwendung Evolutionärer Algorithmen trifft man meist auf restringierte Probleme. Nicht zuletzt deshalb wurde in den vorangegangenen Parameterstudien die Keanesche Bump-Funktion in den Katalog der Testfunktionen aufgenommen. Im folgenden Abschnitt wird die Art der Restriktionsbehandlung nach Hoffmeister und Sprave (1996), die bei den Experimenten verwendet wurde, beschrieben. Weiterhin werden Studien vorgestellt, die die Eignung dieser Form der Restriktionsbehandlung in rangbasierten Optimieralgorithmen aufzeigen.

Formal bedeutet restringierte Optimierung (constraint optimization): Minimiere eine Funktion $f(\vec{x})$ mit

$$
\begin{aligned}
g_i(\vec{x}) &\geq 0 \qquad ; \forall i \in \{1, \ldots, m\} && (8.1)\\
h_j(\vec{x}) &= 0 \qquad ; \forall j \in \{1, \ldots, l\} && (8.2)\\
\vec{x} &= (x_1, \ldots, x_n) \in \mathbb{R}^n && (8.3)\\
f &: M \to \mathbb{R},\ M \subseteq \mathbb{R}^n && (8.4)\\
g_i &: \mathbb{R}^n \mapsto \mathbb{R} && (8.5)\\
h_j &: \mathbb{R}^n \mapsto \mathbb{R}\,. && (8.6)
\end{aligned}
$$

Dabei werden die $g_i(\vec{x})$ *Ungleichheitsrestriktionen* und die $h_j(\vec{x})$ *Gleichheitsrestriktionen* genannt. Ein Testpunkt $\vec{x}$, der (8.2) und (8.3) genügt, wird *gültig*, genannt, ansonsten *ungültig*. Die Menge aller gültigen Punkte wird auch *zulässiges Gebiet* genannt.

Da die meisten klassischen Optimierverfahren aufgrund ihrer inneren Modellannahmen, etwa Stetigkeit oder Differenzierbarkeit, nicht für restringierte Optimierung geeignet sind, sind in den vergangenen Jahrzehnten eine Reihe von Ideen zur Konstruktion entsprechender Ersatzfunktionen entstanden. Diese Straffunktionen (penalty functions) ersetzen die Zielfunktion $f : M \mapsto \mathbb{R}$ durch eine Funktion $F : \mathbb{R}^n \mapsto \mathbb{R}$, für welche im Idealfall gilt:

$$\begin{aligned} \forall x, y \quad &\in \quad M : && (8.7) \\ f(\vec{x}) \le f(\vec{y}) \quad &\Rightarrow \quad F(\vec{x}) \le F(\vec{y}), && (8.8) \end{aligned}$$

so daß sichergestellt ist, daß das Optimum der Ersatzfunktion auch das Optimum der Originalfunktion ist. Weiterhin sollte jeder ungültige Punkt schlechter bewertet werden als jeder gültige, formal:

$$\begin{aligned} \forall x \in M \quad &\wedge \quad \forall y \in \mathbb{R}^n \setminus M && (8.9) \\ F(\vec{x}) \quad &\le \quad F(\vec{y}). && (8.10) \end{aligned}$$

Einige direkte Suchverfahren, zum Beispiel die Rosenbrock-Strategie (Rosenbrock 1960), integrieren sogenannte *partielle Straffunktionen* in ihre Auswertungsroutine. Partielle Straffunktionen werden nur an der Grenze zum gültigen Bereich wirksam:

$$F(\vec{x}) = \begin{cases} f(\vec{x}) & ;\text{if } g_i(\vec{x}) \ge \epsilon \\ p(\vec{x}) & ;\text{else} \end{cases} \tag{8.11}$$

Dabei ist ϵ ein sensibler Parameter: Ein zu großer Wert kann ein Optimum verstecken, das nahe an der Grenze zum unzulässigen Bereich liegt. Ein zu kleiner Wert bestraft ungültige Punkte nicht ausreichend und erzeugt möglicherweise Scheinoptima, die nicht im zulässigen Gebiet liegen.

Idealerweise wäre ϵ gleich Null und die Ersatzfunktion würde der folgenden Bedingung genügen:

$$F(\vec{x}) < F(\vec{x}') \quad ; \forall \vec{x} \in M, \vec{x}' \notin M \tag{8.12}$$

Allerdings kann eine solche Ersatzfunktion in der Regel keinen stetigen oder gar differenzierbaren Übergang zwischen zulässigem und unzulässigem Gebiet garantieren. Nun wird aber bei rangbasierten Suchverfahren, also Verfahren, die nur den Vergleich von Zielfunktionswerten zur Erkundung des Suchraums nutzen, gar keine derartige Anforderung an die Ersatzfunktion gestellt.

Es läßt sich also nur historisch erklären, warum die meisten Ansätze für Restriktionsbehandlungen in rangbasierten direkten Optimierverfahren davon ausgehen, daß eine glatte Ersatzfunktion definiert werden muß. Insbesondere im Bereich der Evolutionären Algorithmen kann, mit Ausnahme der proportionalen Selektion, immer eine ideale Straffunktion konstruiert werden.

Dies geschieht über eine Ersatzfunktion F, die die Qualität eines Suchpunktes nicht in die reellen Zahlen, sondern auf die Menge $\mathbb{B} \times \mathbb{R}$ abbildet.

Def. 11 Sei $f : \mathbb{R}^n \mapsto \mathbb{R}$ eine Zielfunktion mit Restriktionen und zulässigem Gebiet $M \subset \mathbb{R}^n$. Weiterhin sei $d_c : M \mapsto R$ eine Funktion, die den Abstand zum nächsten gültigen Suchpunkt bezogen auf eine Norm liefert. Dann sei

$$F \ : \ \mathbb{R}^n \to \mathbb{B} \times \mathbb{R} \tag{8.13}$$

$$F(\vec{x}) = \begin{cases} (0, f(x)) & ;\text{if } \vec{x} \in M \\ (1, d_c(\vec{x})) & ;\text{else} \end{cases} \tag{8.14}$$

Dann heißt die Funktion d_c *metrische Straffunktion* (engl. metric penalty function, MPF).

Seien nun $x'_1 = (b_1, x_1)$ und $x'_2 = (b_2, y_2)$ beliebig aus $\mathbb{B} \times \mathbb{R}$. Diese kann man nun mittels folgender Ordnungsrelation miteinander vergleichen:

$$\trianglelefteq \ \subseteq \ (\mathbb{B} \times \mathbb{R}) \times (\mathbb{B} \times \mathbb{R}) \tag{8.15}$$

$$x'_1 \trianglelefteq x'_2 \ \Leftrightarrow \ (b_1 < b_2) \vee (b_1 = b_2) \wedge (y_1 \leq y_2) \tag{8.16}$$

Ein direktes, rangbasiertes Optimierverfahren kann nun Restriktionsbehandlung mittels MPF umgestellt werden, indem man als Zielfunktion die Ersatzfunktion F verwendet und als Vergleichsoperator $\trianglelefteq$. Es verbleibt die Aufstellung der Funktion d_c. In vielen Fällen geben bereits die Werte der Restriktionsfunktionen den Grad der jeweiligen Restriktionsverletzungen an. In diesem Fall wählt man zum Beispiel mit

$$d_c(\vec{x}) = \sqrt{\sum_{i=0}^{m} H(-g_i(\vec{x}))g_i(\vec{x})^2} \tag{8.17}$$

die Wurzel aus der Quadratsumme der Verletzungen, wobei H die Heavyside-Funktion ist:

$$H(y) = \begin{cases} 1 & : \quad y > 0 \\ 0 & : \quad y \leq 0 \end{cases} \tag{8.18}$$

Falls aber die Restriktionsfunktionen im Falle der Verletzung kein Maß für den Abstand zum zulässigen Gebiet liefern, kann man im Falle mehrerer Restriktionen

$$d_c(\vec{x}) = \sum_{i=0}^{m} H(-g(\vec{x})) \tag{8.19}$$

wählen und bewertet somit die Anzahl der verletzten Restriktionen. Gibt es schließlich nur eine Restriktion, deren Funktionswert im Falle einer Verletzung kein Maß für den Grad der Verletzung darstellt, so wählt man zum Beispiel

$$d_c(\vec{x}) \quad = \quad d(\vec{x}, \vec{x}^{+}) \tag{8.20}$$

wobei d der Euklidische Abstand im $\mathbb{R}^n$ ist und $\vec{x}^{+}$ der bisher beste gültige Punkt. Dies setzt natürlich voraus, daß überhaupt erst einmal ein gültiger Punkt gefunden wurde — unter der angegebenen Prämisse, daß die Restriktionsfunktion keine Lenkung in Richtung des zulässigen Gebiets bewirkt, ist der Einsatz eines direkten Suchverfahrens ohnehin nur mit einem gültigen Startpunkt sinnvoll.

Die Wahl des besten gültigen Punktes beruht darauf, daß man diesen bei allen Verfahren zur Verfügung hat. Alternativ könnte man auch den räumlich nächsten gültigen Punkt der letzten Iteration wählen, um die Suche nicht künstlich auf einen Bereich zu verengen. Genauere Untersuchungen hierzu sprengen den Rahmen dieser Arbeit. Stattdessen sollen die beiden ersten Ansätze aus den Gleichungen 8.17 und 8.19 im folgenden anhand einer klassischen Testfunktion für die Restriktionsbehandlung in Evolutionsstrategien bewertet werden. Zum Vergleich wurde die in

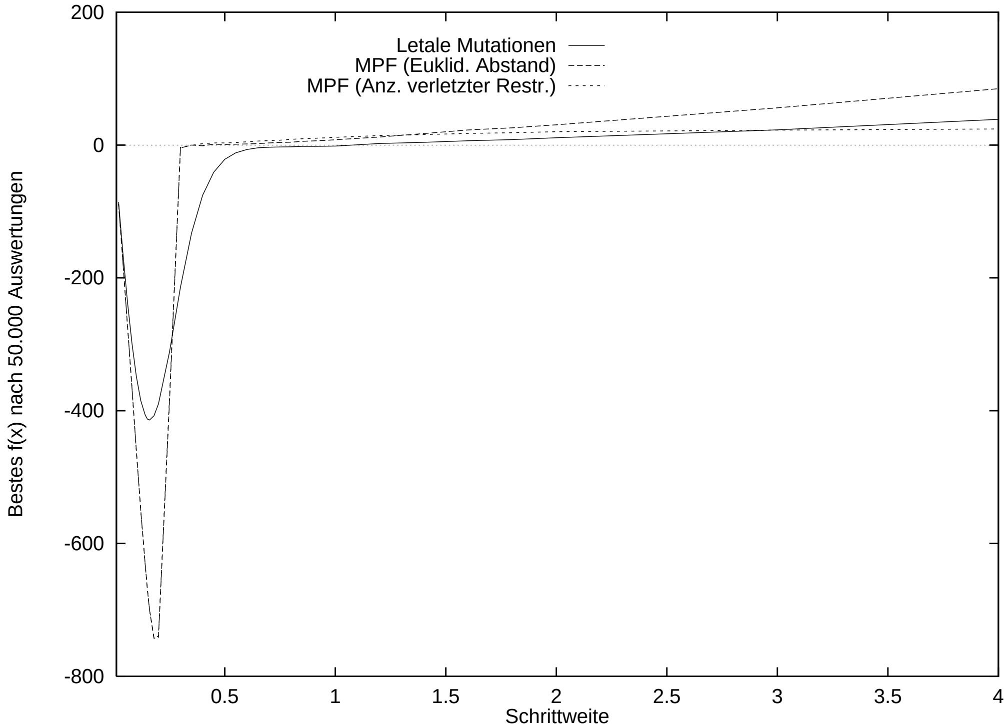

Abbildung 8.1: Korridormodell mit festen Schrittweiten. Im Intervall $[0, 0.5]$ sind die Kurven der MPF-Varianten nahezu deckungsgleich.

der klassischen ES bereits verwendete Methode der Restriktionsbehandlung — ungültige Punkte werden verworfen — herangezogen. Schwefel (1995, S. 134ff.) verwendet eine einfache, beschränkte Funktion zur Analyse der Fortschrittsgeschwindigkeit von Evolutionsstrategien, das sogenannte *Korridormodell*:

$$\begin{aligned} f(\vec{x}) &= -x_1 \\ \text{mit} \quad g_j(\vec{x}) &= b - |x_j| \quad \geq 0 \end{aligned} \tag{8.21}$$

Man erhält einen rechtwinkligen Korridor im n-dimensionalen Raum, der entlang der x_1-Achse linear abfällt. Das Optimum liegt bei $x_1 = -\infty, |x_i| < b, i = 2, \ldots, n$. Gemessen wird bei dieser Zielfunktion der Weg (auf der x_1-Achse), den ein Verfahren mit einer vorgegebenen Zahl von Zielfunktionsauswertungen zurücklegt.

Es wurden zwei Testreihen mit einer $(15, 100)$-ES und $n = 30$ durchgeführt. Jeder Lauf wurde nach 50.000 Auswertungen beendet und der Fortschritt auf der x_1-Achse des bisher besten gültigen Suchpunkts wurde bewertet. Dabei wurde bei ungültigen Punkten die Berechnung der Restriktionen ebenfalls als Auswertung gezählt.

Die erste Studie wurde mit festen Schrittweiten durchgeführt, um zu überprüfen, ob die theoretisch optimale Fortschrittsgeschwindigkeit auch in der Kombination mit MPF gilt.

Rechenberg (1973) gibt diese mit

$$\sigma_{\text{opt}} = b\frac{\sqrt{2\pi}}{n} \tag{8.22}$$

an, wobei b die Korridorbreite aus Gl. 8.21 ist. Für $b = 1$ und $n = 30$ erhalten wir einen Wert von ungefähr 0.08. Abbildung 8.1 zeigt den Zusammenhang zwischen den Schrittweiten und dem erreichten Fortschritt nach 50000 Auswertungen. Offensichtlich liegt die Schrittweite mit dem größten Fortschritt für alle Varianten mit 0.18 ein wenig über der theoretischen Abschätzung. Dies ist jedoch auch plausibel, da diese für eine $(1 + 1)$-ES ermittelt wurde. Der höhere Selektionsdruck einer $(15, 100)$-Strategie scheint eine riskantere Plazierung von Suchpunkten zu erlauben.

Der mittels MPF erreichte Fortschritt (Abb. 8.1) ist deutlich größer als derjenige, der durch Verwerfen ungültiger Punkte erzielt wurde. Besonders erstaunlich ist aber, daß die MPF-Variante, die nur die Anzahl der verletzten Restriktionen als Maß für den Grad der Restriktionsverletzung bewertet, nahezu den gleichen Fortschritt erzielt wie die Quadratsummenminimierung.

Als weiteres Ergebnis läßt sich ablesen, daß die Restriktionsbehandlung mittels MPF sensibler auf Fehleinstellungen der Mutationsschrittweite reagiert.

Für die zweite Testreihe wurde dann die Selbstanpassung nach Schwefel (1995) auf eine gemeinsame Schrittweite für alle x_i eines Suchpunktes (Individuums) angewendet. Alle Individuen wurde mit derselben Schrittweite initialisiert. Abbildung 8.2 zeigt den Fortschritt in Abhängigkeit von der Anfangsschrittweite. Bei

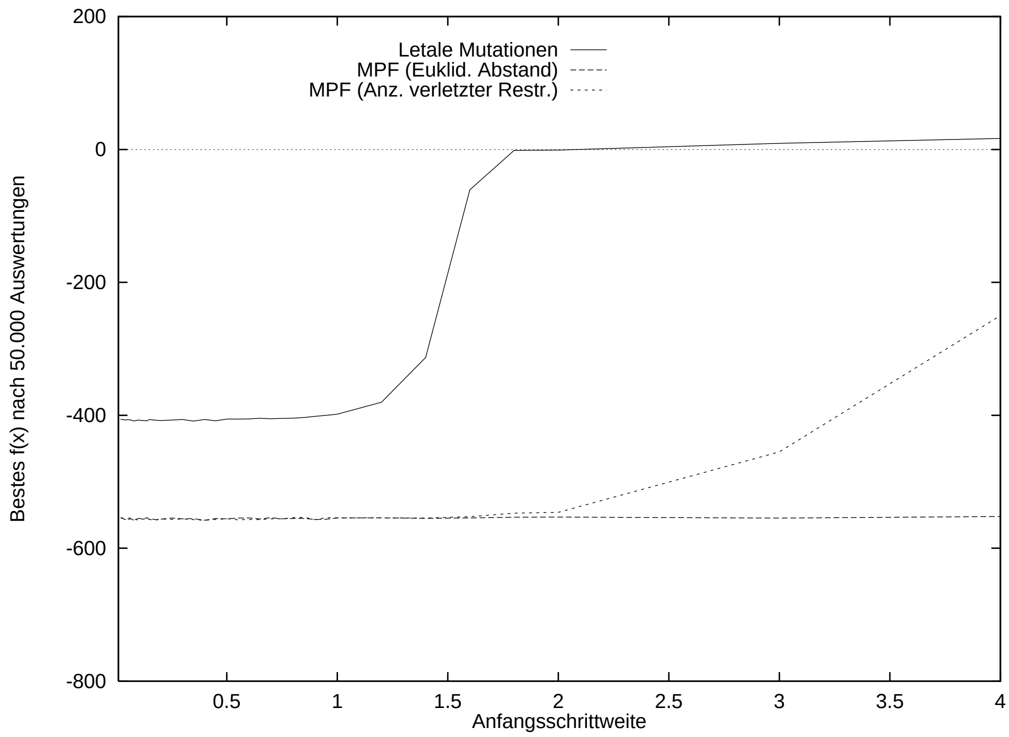

Abbildung 8.2: Korridormodell mit variablen Schrittweiten.

einer Anfangsschrittweite über der zweifachen Korridorbreite schafft es die klassische Form der Restriktionsbehandlung nicht, innerhalb der ersten 50000 Auswertungen überhaupt Fortschritt zu erzielen. Mit kleineren Anfangsschrittweiten liegt der Fortschritt immer noch deutlich unter dem der MPF-Varianten. Wiederum zeigt sich, daß die Information über die Anzahl der verletzten Restriktionen in schwierigen Situationen (hier: zu große Schrittweiten) gegenüber der Verwerfung ungültiger Lösungen von Vorteil sein kann.

8.2 Inhomogene Parallelisierung

In den vorangegangenen Kapiteln wurde die Parallelisierung Evolutionärer Algorithmen immer homogen vorgenommen, alle Teilpopulationen verwendeten dieselben Algorithmen mit derselben Parametrisierung. Dies war im Sinne einer Analyse der Populationstrukturen auch sicherlich sinnvoll. Bei der Anwendung von EA kann es dagegen vorteilhaft sein, Teilpopulationen mit verschiedenen Strategien auszustatten und diese miteinander kommunizieren zu lassen.

Im einfachsten Fall kann eine inhomogene Parallelisierung dadurch erreicht werden, daß zwar alle Teilpopulationen dieselbe Strategie verwenden, diese aber bezüglich eines externen Verfahrensparameters jeweils anders parametrisiert wird. Dies kann als Alternative zur Selbstanpassung dieses Parameters angesehen werden. Bei der in Abschnitt 9.2 beschriebenen Optimierung optischer Mehrschichtsysteme wurde auf diese Art parallel mit einem Satz von 15 verschiedenen Mutationsraten für den ganzzahligen Anteil der internen Problemrepräsentation gearbeitet. Dieses *Pooling* der Mutationsrate kann dabei als Alternative zur Selbstanpassung von Strategieparametern dienen, da immer mehrere Parametereinstellungen zur Verfügung stehen, von denen dann die Teilpopulation mit der besten Einstellung auch den größten Fortschritt erzielt. Zusätzliche bewahren Teilpopulationen mit großen Mutationsraten den explorativen Charakter des Optimierverfahrens. Sie liefern die Vorschläge für potentiell attraktive Gebiete des Suchraums, die von den Teilpopulationen mit geringer Mutationsrate lokal optimiert werden können.

8.3 Lokale adaptive Akzeptanzschwellen

Die Idee einer Akzeptanzschwelle für probabilistische Suchverfahren stellt bei Dueck, Scheuer und Wallmeier (1993) das wesentliche Element ihres *Sintflutalgorithmus'* dar. Das Verfahren besteht aus einem Variationsschritt für Punkte im Suchraum und einer Regel für die Anpassung einer Qualtitätsschwelle, die nicht unterschritten werden darf. Die Variation kann dabei ungerichtet (z.B. Random Walk) sein, oder sie kann auf Heuristiken für die Klasse der zu lösenden Problemen basieren (z.B. k-opt beim Traveling-Salesperson-Problem). Der Schwellwert wird im Laufe der Iteration monoton angehoben. Variationen von Suchpunkten werden akzeptiert, wenn ihr Zielfunktionswert über dem Schwellwert liegt, ansonsten werden sie verworfen. Auf diese Weise läßt das Verfahren in begrenztem

Maße Verschlechterungen zu. Der Vergleich mit einer ansteigenden Flut, bei der nur Lebewesen überleben, die immer rechtzeitig an Höhe gewinnen, führte zum Namen dieses Verfahrens.

Dieser Ansatz läßt sich leicht mit Evolutionären Algorithmen kombinieren, da man bereits mit der Mutation einen Variationsoperator gegeben hat. Die Einführung eines Schwellwerts bedeutet eine Veränderung der Selektion. Während Dueck, Scheuer und Wallmeier (1993) den Schwellwert nur von der Iterationszahl abhängig machen, schlagen Rudolph und Sprave (1995) eine adaptive Anpassung des Schwellwerts in Verbindung mit einem Genetischen Algorithmus und einem Nachbarschaftsmodell vor. Hierzu wurde ein zeitliches Evolutionsfenster der Länge δ eingeführt, das die Anzahl von Generationen vorgibt, in denen die Population einen Schwellwert übertreffen[1] muß. Der neue Schwellwert wird dann aber nicht anhand der Iterationszahl festgesetzt, sondern auf die beste, vor δ Generationen erreichte Qualität gesetzt, wenn diese über dem aktuellen Schwellwert liegt, ansonsten wird dieser beibehalten, um Monotonie sicherzustellen. Im Nachbarschaftsmodell erhält dann auch noch jede Position k in der Population ihren eigenen Schwellwert $\tau_k^{(t)}$ mit:

$$\tau_k^{(t)} := \begin{cases} f(x_k^{(0)}) & \text{, if } t < \delta \\ \max\{\tau_k^{(t-1)}, f(x_k^{(t-\delta)})\} & \text{, sonst.} \end{cases} \tag{8.23}$$

Von Rudolph und Sprave (1995) wurde anhand einer Parameterstudie der Einfluß der Länge des zeitlichen Evolutionsfensters und der Nachbarschaftsgröße in einem ringförmigen Nachbarschaftsmodell mit zusammenhängenden Nachbarschaften untersucht. Für eine zweite Studie (Rudolph und Sprave 1996) wurden die gleichen Experimente auf einem Torus durchgeführt, um zusätzlich den Einfluß der Verbindungstopologie zu untersuchen. Dabei wurden die Ergebnisse gegen die Nachbarschaftsgröße aufgetragen. Um auf einem zweidimensionalen Gitter symmetrische, zusammenhängende Nachbarschaften zu erhalten, kann man sich der in der Bildverarbeitung gebräuchlichen *Chamfer-Distanz* (Borgefors 1984a; Borgefors 1984b) bedienen, die eine relativ glatte Transformation des Euklidischen Abstands auf rechtwinklige Gitter darstellt. Die Chamfer-Distanzen bis

[1] Verwendet wurde lokale proportionale Selektion gemäß Abschnitt 5.3.1. Daher wird in diesem Abschnitt von Maximierungsaufgaben ausgegangen.

zum Abstand 9 lassen sich aus folgender Matrix C ablesen:

$$C = \begin{pmatrix} 9 & 8 & 7 & 6 & 7 & 8 & 9 \\ 8 & 5 & 4 & 3 & 4 & 5 & 8 \\ 7 & 4 & 2 & 1 & 2 & 4 & 7 \\ 6 & 3 & 1 & 0 & 1 & 3 & 6 \\ 7 & 4 & 2 & 1 & 2 & 4 & 7 \\ 8 & 5 & 4 & 3 & 4 & 5 & 8 \\ 9 & 8 & 7 & 6 & 7 & 8 & 9 \end{pmatrix} \tag{8.24}$$

Die Matrixelemente enthalten die Chamfer-Distanz zum mittleren Matrixelement. Diese Matrix kann man als Maske auf einem Torus verwenden, um die Chamfer-Abstände zwischen beliebigen Gitterpunkten zu ermitteln. Als Nachbarschaft eines bestimmten Individuums verwendet man also alle Individuen mit einem vorgegebenen maximalen Chamfer-Abstand. Man gelangt so zu folgendem Populationsmodell für einen $w \times h$-Torus mit einer k-Chamfer-Nachbarschaft:

$$\begin{aligned} \Pi_{\text{chamfer}} &= (X, \mathcal{E}, \mathcal{Q}) \\ X &= (0, \ldots, \lambda - 1), \lambda = w \cdot h, w, h \in \lambda\mathbb{Z} \\ Q &= (\{x_0\}, \ldots, \{x_{\lambda-1}\}) \\ \mathcal{E} &= (E_0, \ldots, E_{\lambda-1}) \\ E_i &= \{j \in X | c_{i,j} \leq k\} \end{aligned} \tag{8.25}$$

Dabei sei $c_{i,j}$ der Abstand der Individuen i und j, der sich aus der Anwendung der obigen Matrix auf einen irgendwie durchnumerierten $w \times h$-Torus ergibt. Im Experiment wurden die maximalen Chamfer-Distanzen $k = 1, \ldots, 9, 14, 20, 27, 35$ betrachtet. Damit erhält man die Nachbarschaftsgrößen $5, 9, 13, 21, 25, 29, 37, 45, 49, 81, 121, 169$ und 225.

Die Experimente wurden auf zwei Zielfunktionen durchgeführt: einem pseudo-Booleschen und einem kontinuierlichen Problem. Die erste Funktion ist ein multiples Rucksackproblem, das von Khuri, Bäck und Heitkötter (1994) mit einem panmiktischen GA untersucht wurde, so daß Vergleichsdaten existieren. Formal läßt sich dieses Problem schreiben als

$$f(x) = c^\top x \to \max! \tag{8.26}$$

$$\text{mit } Bx \leq b \tag{8.27}$$

und mit $x \in \mathbb{B}^\ell$, $c \in \mathbb{R}_+^\ell$, $b \in \mathbb{R}_+^m$ und $B \in \mathbb{R}_+^{m \times \ell}$. Die Restriktionen wurden mit derselben Straffunktion in die Zielfunktion eingebracht wie bei Khuri, Bäck und Heitkötter (1994):

$$f_{knap}(x) = c^\top x - \beta \cdot c_{max} \to \max! \tag{8.28}$$

wobei β die Anzahl der verletzten Restriktionen angibt und c_{max} den größten Wert im Kostenvektor c.

Als zweite Zielfunktion wurde die modifizierte Rastrigin-Funktion aus Gl. 7.7 verwendet. Um eine Maximierungsaufgabe mit positiver Fitneß zu erhalten, wurde der Zielfunktionswert von 5000 abgezogen. Verwendet wurde eine Dimension von $n = 20$ mit den Parametern $A = 10$ und $\omega = 2\pi$. Als interne Repräsentation wurde für jedes x_i jeweils ein Bitstring der Länge 20 verwendet und als Gray-Code-Festkommadarstellung interpretiert. Für beide Probleme wurden der Nachbarschaftsradius und die Verzögerung der Schwellwertanpassung variiert. Verglichen wurden die Ergebnisse auf dem Torus mit den Ergebnissen, die mit denselben Zielfunktionen in einem ringförmigen Nachbarschaftsmodell erzielt wurden (Rudolph und Sprave 1995). Die Erfolgsraten wurden aus 100 Läufen gemittelt.

Bei dem multiplen Rucksackproblem zeigt sich, daß das ringförmige Nachbarschaftsmodell (Abb. 8.3) robuster auf die Veränderung der beiden Verfahrensparameter reagiert, allerdings auch nicht ganz an die beste Erfolgsrate des Chamfer-Modells (Abb. 8.4) herankommt. Die relativ große Nachbarschaft von 40, die zum Erreichen der besten Erfolgsrate von ca. 0.85 benötigt wird, bedeutet einen hohen Kommunikationsbedarf im Falle einer parallelen Implementierung. Das Chamfer-Modell hingegen erzielt mit der kleinsten möglichen Nachbarschaftsgröße von 5 die beste Erfolgsrate von ca. 0.95 und ist somit, bezogen auf die hier verwendete GA-Variante, auch bei geringer Kommunikationsbandbreite noch effizient parallelisierbar. Allerdings reagiert das Chamfer-Modell auch empfindlicher auf Fehleinstellungen der Nachbarschaftsgröße.

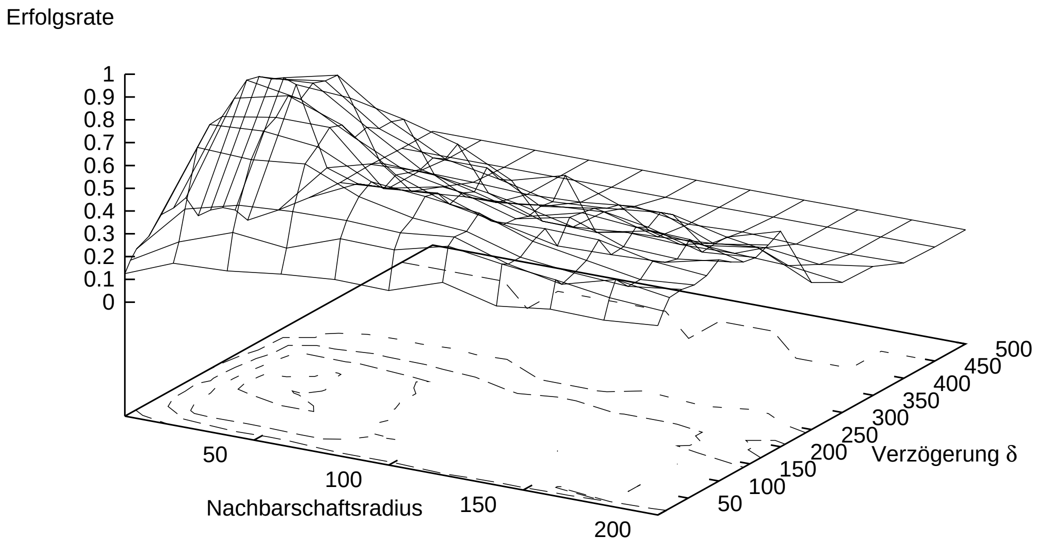

Abbildung 8.3: Erfolgsrate des GA mit Schwellwertanpasung in einem ringförmigen Nachbarschaftsmodell. Die höchste Erfolgsrate wurde mit einer Nachbarschaftsgröße von 40 und einer Schwellwertverzögerung von 100 Generationen erzielt.

Beide Topologien erzielen mit relativ großen Schwellwertverzögerungen (Ring: 100, Torus: 140) die besten Ergebnisse. Da aber ohne Schwellwertanpassung innerhalb von 500 Generation das Optimum überhaupt nicht gefunden wurde (dies ist konsistent mit den Ergebnissen von Khuri, Bäck und Heitkötter (1994)), kann man davon ausgehen, daß bereits eine schwache elitäre Komponente, eben der monoton steigende Schwellwert mit langer Verzögerung, genügt, um lokale proportionale Selektion für dieses Problem zu beschleunigen.

Bei der Rastrigin-Funktion stellt sich das ringförmige Nachbarschaftsmodell als äußerst robust gegenüber der Variation beider Parameter dar. In Abb. 8.5 ist der mittlere beste Zielfunktionswert aus 100 Läufen aufgetragen. Wieder sind die besten Werte des Chamfer-Modells (Abb. 8.6) geringfügig besser, allerdings nur für extrem kleine Werte mindestens eines der beiden variierten Parameter. Große Nachbarschaften und insbesondere lange Schwellwertverzögerungen führen zu im Schnitt weit vom Optimum entfernten Zielfunktionswerten.

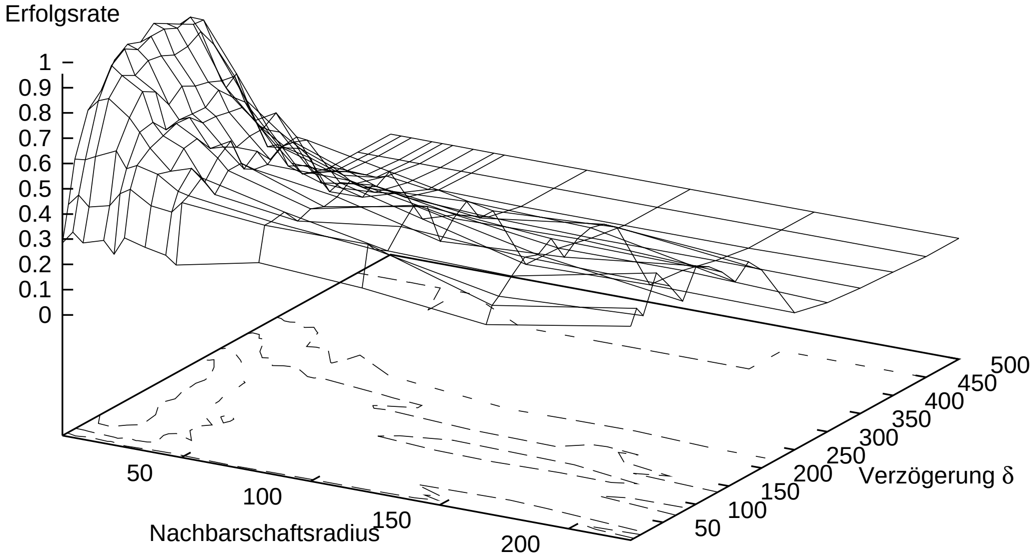

Abbildung 8.4: Erfolgsrate des GA mit Schwellwertanpassung in einem Torus mit Chamfer-Nachbarschaft. Die höchste Erfolgsrate wurde mit einer Nachbarschaftsgröße von 5 und einer Schwellwertverzögerung von 140 Generationen erzielt.

Einen möglichen Erklärungsansatz liefern die probabilistischen Durchmesser der beiden Populationsstrukturen. In Abb. 8.7 ist der probabilistische Durchmesser einer Ring- und einer Torustopologie zu verschiedenen Nachbarschaftsgrößen und für ein Sättigungsniveau von 0.999 $(\varepsilon = 0.001)$ und $2/3$ $(\varepsilon = 1/3)$ aufgetragen. Als Selektionsoperator wurde lineare Rangselektion mit $\alpha = 1,5$ verwendet. Als Populationsgröße wurde 625 (25×25 beim Torus) gewählt. Interessanterweise fallen die Werte für die kleinsten Nachbarschaften beim Torus mit denen für mittlere Nachbarschaften beim Ring zusammen.

Die Verwendung adaptiver Schwellwerte ist ein vielversprechender Kompromiß zwischen elitärer und nichtelitärer Selektion. Die Untersuchung dieses Ansatzes in Verbindung mit strukturierten Populationen hat jedoch ergeben, daß sich die Effekte aus lokaler Selektion und Schwellwertanpassung nicht voneinander trennen lassen. Daher sollte die Schwellwertanpassung zunächst im panmiktischen Fall untersucht werden und dort mit dem verwandten Ansatz der Alterung von Individuen verglichen werden.

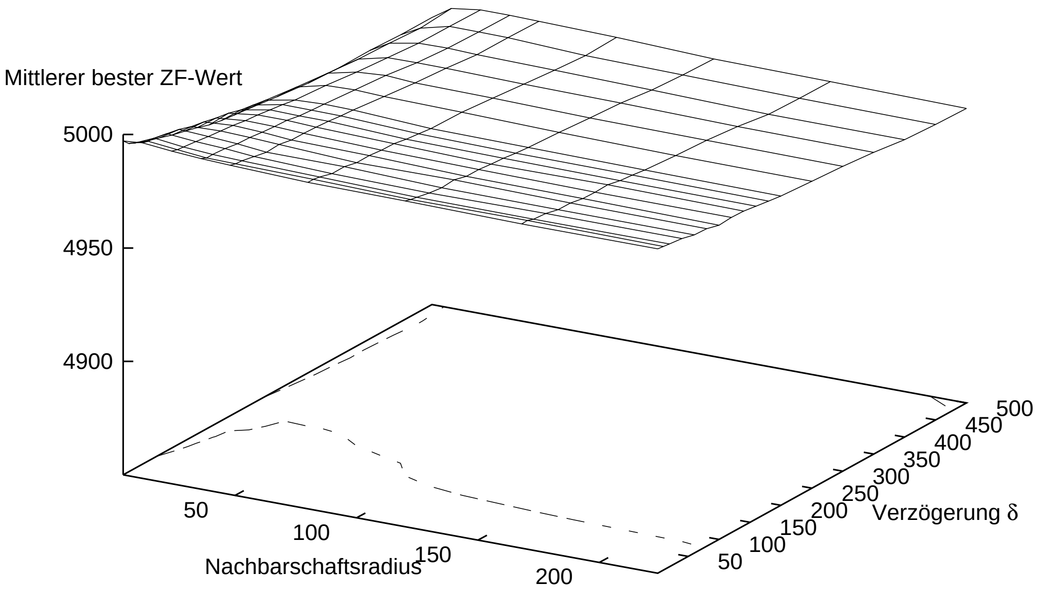

Abbildung 8.5: Mittlerer bester Zielfunktionswert des GA mit Schwellwertanpassung in einem ringförmigen Nachbarschaftsmodell auf der Rastrigin-Funktion.

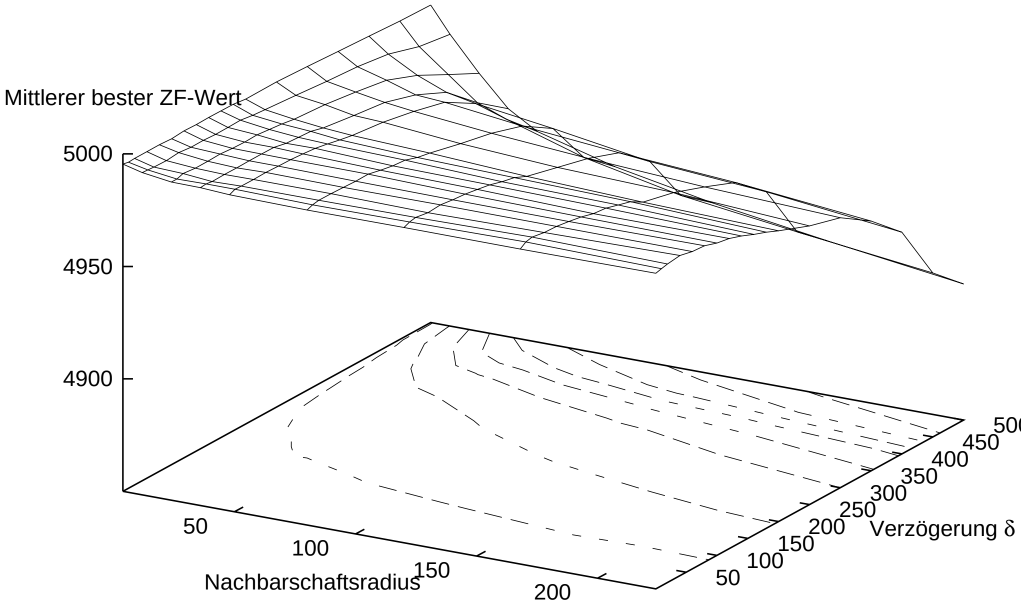

Abbildung 8.6: Mittlerer bester Zielfunktionswert des GA mit Schwellwertanpassung in einem Torus mit Chamfer-Nachbarschaft auf der Rastrigin-Funktion.

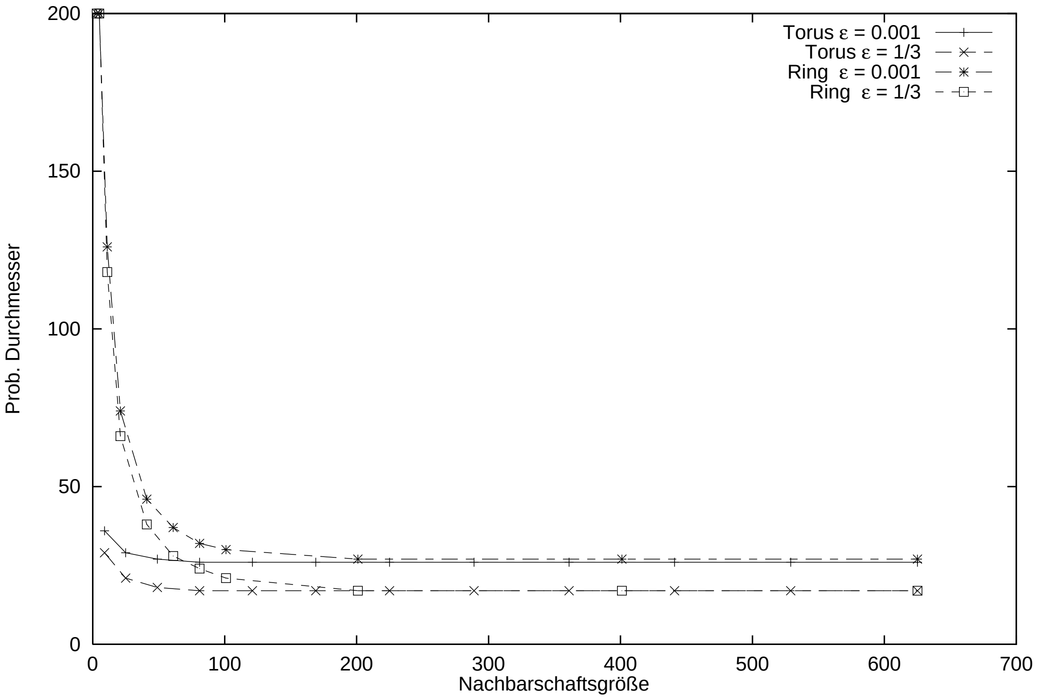

Abbildung 8.7: Probabilistischer Durchmesser eines ringförmigen und eines torusförmigen Nachbarschaftsmodells mit linearer Rangselektion, aufgetragen über die Nachbarschaftsgröße.

Kapitel 9

Das PACE-Populationsmodell und seine Anwendung

9.1 Das PACE-Populationsmodell

Ein Populationsmodell, das sich im Hinblick auf die Parallelisierung und die Ergebnisqualität als besonders anwendungstauglich erwiesen hat, ist das PACE-Populationsmodell. PACE steht dabei für *PArallel Cellular Evolutionary algorithm* und stellt eine Erweiterung des Modells von Sprave (1990) dar. Das Populationsmodell ist eine Kombination aus einem Pollinationsmodell und einem gitterbasierten Nachbarschaftsmodell. Jeder Prozessor bearbeitet einen Abschnitt eines zum Torus gefalteten rechtwinkligen Gitters, wobei an den Rändern die Nachbarschaft über die Prozessorgrenzen hinaus fortgesetzt wird. Zur Verringerung der Kommunikationslast oder auch zur Senkung der Takeover-Zeit gibt es eine Isolationszeit $\eta \in \mathbb{N}$. Die Ränder werden nur alle η Generationen ausgetauscht, ansonsten werden die lokalen Teilpopulationen ihrerseits zu kleinen Tori gefaltet.

Sei $N \in \mathbb{N}$ die Zahl der Prozessoren und $\lambda = \lambda' N, \lambda \in \mathbb{N}$ die Populationsgröße. Weiterhin seien $w, h \in \mathbb{N}$ die Breite und Höhe eines rechtwinkligen Gitters mit $h = h'N, h' \in \mathbb{N}$, und ρ ein Nachbarschaftsradius analog zu Gl. 4.22. Mit $+_k$ als Additionsoperator in der Restklassengruppe $k\mathbb{Z}$ läßt sich das PACE-

Populationsmodell folgendermaßen aufschreiben:

$$
\begin{aligned}
\Pi_{\text{PACE}} &= (X, \mathcal{E}^{(t)}, \mathcal{Q}) \\
X &= (0, \ldots, \lambda - 1) \\
\mathcal{Q} &= (\{x_0\}, \ldots, \{x_{\lambda-1}\}) \\
\mathcal{E}^{(t)} &= (E_0^t, \ldots, E_{\lambda-1}^t) \\
E_i^t &= \begin{cases} \{w\lfloor i/w \rfloor +_h y) + (i +_w x) | - \rho \leq x, y \leq \rho\} \\ \qquad \text{für } t \equiv 0 \mod \eta \\ \\ \{i \lfloor i/h'w \rfloor + [(i \bmod h'w)/w +_{h'} y] + (i +_w x) | \\ \qquad - \rho \leq x, y \leq \rho\} \text{ sonst .} \end{cases}
\end{aligned}
\tag{9.1}
$$

Die im Grunde einfachen Indexberechnungen bewirken, daß in Isolationszeiten innerhalb der Teilpopulationen modulo der lokalen Gitterhöhe gerechnet wird, also N Tori der Dimension $h' \times w$ entstehen, während alle η Generationen auf einem $h \times w$-Torus gearbeitet wird. In Abb. 9.1 wird das Populationsmodell in beiden Phasen exemplarisch skizziert. Dieses Populationsmodell vereinigt das hohe Maß an Lokalität des klassischen Nachbarschaftsmodells mit der skalierbaren Parallelität des Migrationsmodells. Durch die Wahl der Teilgittergrößen kann die lokale Prozessorlast gesteuert werden. Gitterhöhe und Nachbarschaftsgröße erlauben die Einstellung der Datenmenge bei der Kommunikation, während die Isolationszeit die Kommunikationsfrequenz bestimmt. Ein Evolutionärer Algorithmus mit dieser Populationsstruktur wurde als Benchmark zur Ausmessung des Kommunikationssystems bei der Beschaffung eines Parallelrechners für den Sonderforschungsbereich 531 *Design und Management komplexer technischer Prozesse und Systeme mit Methoden der Computational Intelligence* eingesetzt.

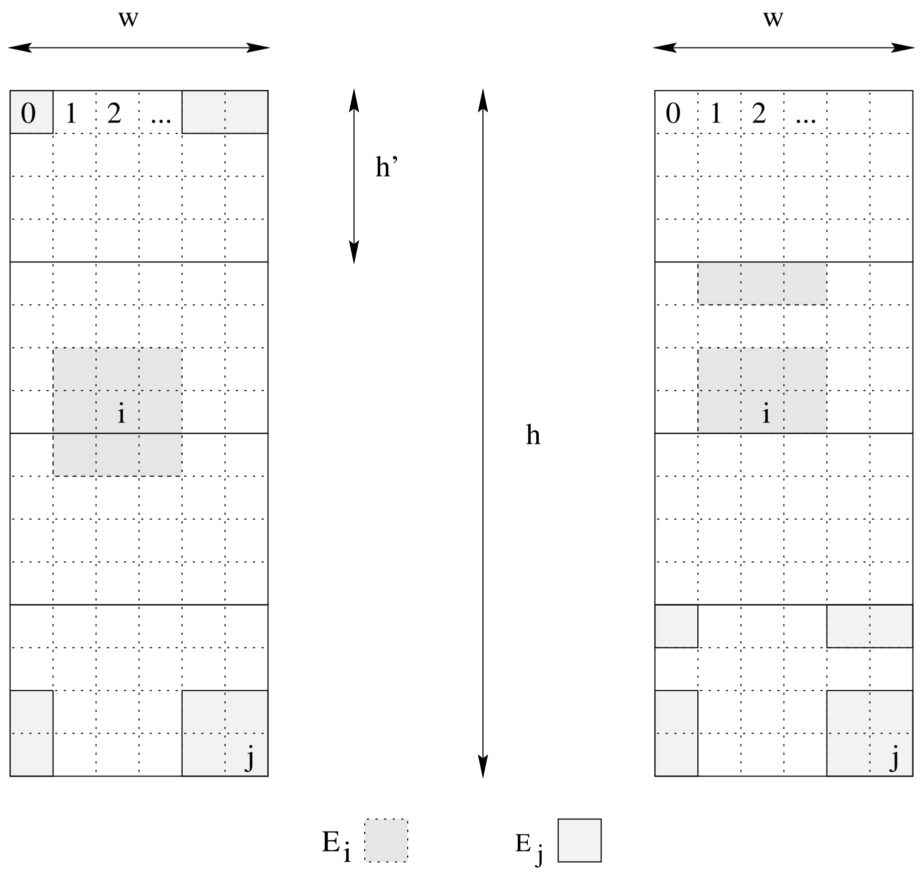

Abbildung 9.1: Schematische Darstellung des PACE-Populationsmodells. Zu zwei gegebenen Indizes i und j sind die Mengen E_i^t und E_j^t schraffiert gezeichnet, links für $t \equiv 0 \ mod \ \eta$, rechts für $t \not\equiv 0 \ mod \ \eta$.

9.2 Design optischer Mehrschichtsysteme

9.2.1 Problemstellung

Bei der Konstruktion optischer Filter ist es das Ziel, einen durch die spätere Anwendung vorgegebenen Bereich von Wellenlängen möglichst exakt zu isolieren. Der Frequenzverlauf eines solchen optischen Systems ist also idealerweise rechteckig. Auch wenn dies physikalisch nicht möglich ist, so kann man doch durch Aufdampfen einer Vielzahl von Schichten mit unterschiedlichen Brechungsindizes sehr gute Annäherungen erzielen.

Das physikalische Modell optischer Mehrschichtsysteme (multi-layer optical coatings – MOCs) wird in der Literatur (Tikhonravov und Dobrowolski 1993) beschrieben als Folge planparalleler, isotroper Schichten, die an zwei homogene, isotrope Medien angrenzen, in der Regel Luft auf der Eintrittsseite und ein Substrat (z.B. Glas) auf der Austrittsseite des Lichts. Ein MOC kann formal als Paar von Vektoren beschrieben werden, und zwar dem Vektor der Schichtdicken $\vec{d} = (d_1, \ldots, d_n)$ und dem der korrespondierenden Brechungsindizes $\vec{\eta} = (\eta_1, \ldots, \eta_n)$. Aus diesen Vektoren kann ein Reflexionsprofil $R(\vec{d}, \vec{\eta}, \lambda)$ berechnet werden, das zu jeder Wellenlänge λ die Reflexion in Prozent liefert (Furman und Tikhonravov 1992). Gemäß Bäck und Schütz (1995) wird daraus ein Optimierungsproblem, indem man über $N \in \mathbb{N}$ äquidistant aus dem Intervall $[\lambda_1, \lambda_N]$ ermittelte Stützstellen die Summe der quadrierten Abweichungen zum Zielprofil $\hat{R}(\lambda_i)$ aufsummiert:

$$f_{MOC}(\vec{d}, \vec{\eta}, N) \quad = \quad \sum_{i=1}^{N} \left[R(\vec{d}, \vec{\eta}, \lambda_i) - \hat{R}(\lambda_i) \right]^2 \;\rightarrow \min! \tag{9.2}$$

$$\text{mit } \sum_{i=1}^{N} d_i \eta_i \quad \leq \quad D_{opt} \, . \tag{9.3}$$

Die Summe in Gl. 9.3 wird als *optische Dicke* bezeichnet. Von dieser hängt die erreichbare Qualität eines Filterdesigns ab, allerdings ist der Zusammenhang bisher nur für relativ einfache Probleme untersucht worden. Für das in Abschnitt 9.2.4 beschriebene Referenzproblem gibt es zwei Abschätzungen für den erreichbaren Restfehler in Abhängigkeit von der optischen Dicke (Thikonravov, Trubetskov, Dobrowolski und Sullivan 1995; Willey 1993). In beiden Fällen wird davon ausgegangen, daß mit zunehmender optischer Dicke das Minimum von Gl. 9.2 monoton fällt. Daher ist die Angabe einer maximalen optischen Dicke Bestandteil einer Filterdesignaufgabe.

Da die Brechungsindizes mit einer endlichen Anzahl von zur Verfügung stehenden Schichtmaterialien korrespondieren, erhält man ein gemischt-ganzzahliges, restringiertes Optimierproblem mit variabler Dimension. Dieses Problem fällt damit in eine Klasse, für die es keine Standardverfahren gibt.

9.2.2 Interne Repräsentation

In Arbeiten von Schütz (1996) und Bäck und Schütz (1996) wurde anhand des MOC-Problems die Fähigkeit einer Evolutionsstrategie untersucht, die Problemdimension als Strategieparameter zu behandeln und analog zu den Schrittweiten selbst anzupassen. Dazu wurde neben einem reellen Vektor für die Schichtdicken ein ganzzahliger Vektor für die Schichtmaterialien sowie ein ganzzahliger Wert für die Anzahl der Schichten als Repräsentation gewählt. Der Wert eines Schichtmaterials verweist dabei auf eine Tabelle der Brechungsindizes der zur Verfügung stehenden Materialien. Zur Veränderung der Schichtanzahl wurden Operatoren definiert, die in Anlehnung an ähnliche Veränderungen natürlicher Genome *Gendeletion* und *Genduplikation* genannt wurden.

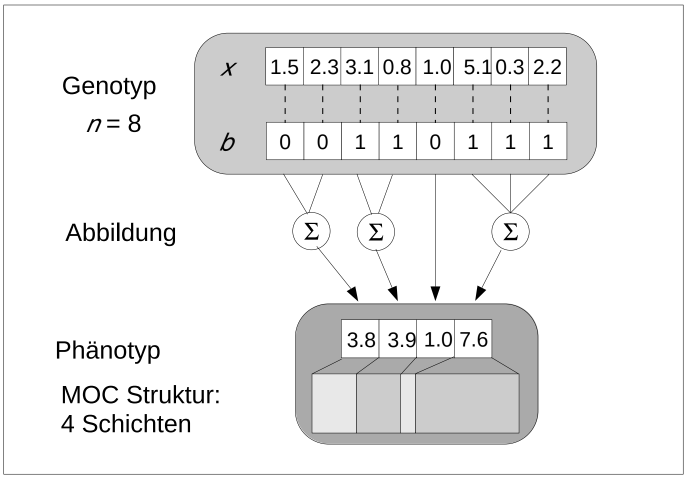

Abbildung 9.2: Abbildung der internen Repräsentation (Genotyp) fester Länge auf eine Filterstruktur (Phänotyp) mit reduzierter Dimension.

Da insbesondere die Selbstanpassung der Problemdimension nicht in der beabsichtigten Weise funktionierte, wurde von Schütz und Sprave (1996) für ein MOC-Designproblem mit nur zwei verschiedenen Schichtmaterialien eine alternative Repräsentation verwendet, die sich auf die beiden Standardrepräsentationen, reellwertiger Vektor und Bit-String, abstützt. Diese Kodierung besitzt eine feste Obergrenze für die Anzahl der Schichten, so daß die Zielfunktion gegenüber dem EA eine feste Dimension aufweist. Für jede mögliche Schicht existiert genau ein reeller Wert für die Dicke der Schicht und ein Bit, das entscheidet, welchen Materialindex diese Schicht erhält. Bei der Auswertung der Zielfunktion findet eine Dimensionsreduktion statt, indem aneinandergrenzende Schichten gleichen Materials zu einer zusammengefaßt werden. Abb. 9.2 skizziert dies exemplarisch. Diese interne Repräsentation kann mit den jeweiligen Standardoperatoren für reelle bzw. binäre Parameter mutiert und rekombiniert werden.

Rolf (1998) hat auf die Bevorzugung mittlerer Dimensionen einer solchen Kodierung hingewiesen. Tatsächlich führt eine gleichverteilt zufällige Belegung des diskreten Anteils einer internen Repräsentation mit Dimension n zur einer Binomialverteilung bei den resultierenden Schichtanzahlen (Sprave und Rolf 1998) mit dem wahrscheinlichsten Wert $n/2+1$ (n ungerade).

9.2.3 Populationsstruktur und Parallelisierung

Panmiktische Versuche bei Schütz (1996) und bei Bäck und Schütz (1996) zeigten zwar die prinzipielle Eignung Evolutionärer Algorithmen für dieses Problem, die Qualität von klassischen Syntheseverfahren in Kombination mit Expertenwissen konnte aber nicht erreicht werden. Schütz und Sprave (1996) verwendeten das PACE-Populationsmodell in Verbindung mit einer inhomogenen Parametrisierung der Teilpopulationen, wie sie in Abschnitt 8.2 vorgestellt wurde. Während die Schichtdicken mittels selbstangepaßter Schrittweiten mutiert wurden, wurde die Mutationsrate für die ganzzahligen Problemparameter innerhalb jeder Teilpopulation auf einen anderen Wert gesetzt.

9.2.4 Ergebnisse

Als Designproblem wurde eine Referenzaufgabe gewählt, die das Ziel setzt, mit zwei Materialien, Germanium und Zinksulfid, eine möglichst gute Antireflexionsbeschichtung für den Infrarotbereich ($7,7\mu m$ — $12,3\mu m$) zu finden. Die besten Ergebnisse wurden erzielt mit einem PACE-Populationsmodell mit $w=30$, $h'=30$, $h=4500$ und $\rho=1$. Als Selektionsoperator erwies sich die Partnerselektion (mating selection) von Sprave (1994) als erfolgreich, also lokale Turnierselektion ohne Zurücklegen, wobei die Turniergröße gleich der Nachbarschaftsgröße ist.

Eine Lauf benötigte dabei in Abhängigkeit von der optischen Dicke Rechenzeit von wenigen Stunden bis zu mehreren Tagen auf einem Rechner der Firma Parsy-Tec (Power-Xplorer), einem Mehrprozessorrechner mit 16 Prozessoren vom Typ Power-PC 601. Die Vielzahl der Experimente führte nicht nur dazu, daß zu den meisten optischen Dicken, zu denen eine gute Lösung bekannt ist, eine vergleichbar gute gefunden wurde, sondern ergab auch eine gute Annäherung an die theoretischen Abschätzungen. Bemerkenswert an diesen Experimenten ist, daß zu einem

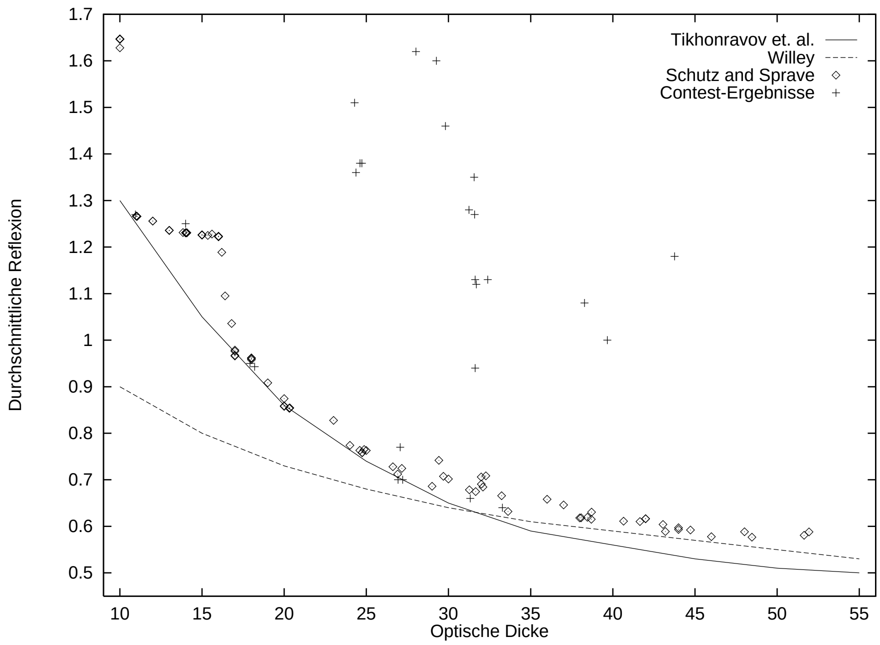

Abbildung 9.3: Ergebnisse der PACE-Optimierung von Schütz und Sprave (1996) im Vergleich zu zwei theoretischen Abschätzungen für die minimal erreichbare Reflexion in Abhängigkeit von der optischen Dicke für ein Referenzproblem mit zwei Brechungsindizes. Zusätzlich sind als gut dokumentierte Ergebnisse anderer Forscher aus einem Contest eingetragen. Aufgetragen ist die Reflexion in Prozent gegen die Schichtdicke in $10^{-6}m$.

Anwendungsproblem, das sowohl empirisch als auch theoretisch Ziel zahlreicher Untersuchungen war und ist, Lösungen erzielt werden konnten, die die meisten mit Expertenwissen erzielten Lösungen übertreffen. Dafür scheint der hohe Rechenaufwand durchaus gerechtfertigt zu sein.

Eine weitere erfolgreiche Anwendung des PACE-Populationsmodells bestand in der Parameterschätzung von ökonomischen Prognosemodellen (Schwefel, Sprave, Preuß, Rolf, Lewandowski und Theis 1997). Die verwendete Prognosemethode (Lewandowski 1980, S. 337 ff.) bestehen aus einer Folge von nichtlinearen, z.T. auch nichtstetigen Transformationen auf multivariaten Einflußzeitreihen, mit Hilfe derer ein explikatives Modell über den Zusammenhang zwischen den Einflußzeitreihen und der Zielzeitreihe erstellt wird. Die Qualität einer Parametrisierung kann man über den Quadratsummenabstand zwischen den Vergangenheitswerten der (empirischen) Zielzeitreihe und dem vom Modell gelieferten Schätzer ermitteln. Praxisrelevante Modelle verfügen über 50 und mehr Parameter. Die simultane Optimierung aller Parameter durch eine parallele Evolutionsstrategie mit dem PACE-Populationsmodell lieferte eine wesentlich bessere Anpassung als die bisher praktizierte getrennte Optimierung einzelner Parameter.

Kapitel 10

Zusammenfassung und Ausblick

Die allgegenwärtige Frage bei einheitlichen Modellierungen ist, ob sie irgendetwas anderes repräsentieren als nur einen Formalismus.

So lautete die Kritik eines Gutachters für die *Conference on Evolutionary Computation* (CEC'99), zu der eine Kurzfassung der Modellierung aus Kap. 4 eingereicht wurde (Sprave 1999b). Nur ein Formalismus? Wenn ein Wissenschaftler mit mathematisch-technischer Ausbildung einen der vielen kürzlich erschienenen Konferenzbände aus dem Bereich der Computational Intelligence durchblättert, wird er nicht glauben können, daß dieses Forschungsgebiet seit über 30 Jahren existiert, so uneinheitlich ist die Terminologie, so unpräzise ist häufig die Darstellung mathematischer Verfahren. Mathematik und Informatik bieten Formalismen, um Algorithmen kompakt und nachvollziehbar aufzuschreiben. Natürlich dürfen Definitionen in Tupelschreibweise und Pseudo-Code nicht alleiniges Beschreibungsmittel sein, vielmehr sollten sie durch erklärenden und motivierenden Text ergänzt werden. Wenn es aber um die Nachvollziehbarkeit von Experimenten geht, wenn der Gültigkeitsbereich theoretischer Aussagen sichergestellt werden soll, so müssen formale Modelle zum Einsatz kommen. Daß diese dann ihre Tauglichkeit unter Beweis stellen müssen, versteht sich von selbst.

Der in dieser Arbeit vorgestellte Modellierungsansatz für Populationsstrukturen in Evolutionären Algorithmen bietet sicher noch Ansatzpunkte für Kritik und Verbesserungen. Dem Argument, *just a formalism* zu sein, kann er jedoch einiges entgegenhalten: Es gibt zur Zeit in der Literatur kein weiteres formales Modell, das fein- und grobkörnige Populationsstrukturen einheitlich behandelt. Dabei wird vor allem auch von Implementierungsdetails abstrahiert, so daß es keine Rolle mehr spielt, ob lokale Selektion mit dem Ziel der Diversitätserhaltung oder der Parallelisierung verwendet wurde. Die Modellierung schließt das klassische, panmiktische Modell in kanonischer Weise mit ein, so daß man die Plausibilität theoretischer Ergebnisse daraufhin überprüfen kann, ob sie im panmiktischen Fall bereits Bekanntes liefern. Diese Eigenschaft hängt eng zusammen mit der Skalierbarkeit dieses Ansatzes, die auf die Dualität von Hypergraph und dualem Hypergraphen zurückzuführen ist. Während in Abschnitt 4.5 und Kap. 9 gezeigt wurde, daß sowohl definierte Populationsstrukturen einfach auf Programmiersprachen abbildbar sind als auch daß relativ komplizierte, existierende Implementierungen nachträglich formalisiert werden können, liefert Kap. 6 ein Beispiel dafür, daß man in diesem Formalismus rechnen und mit ihm zu theoretischen Aussagen gelangen kann.

Die Frage nach der Wahl der richtigen Populationsstruktur kann mit dieser Arbeit sicher nicht beantwortet werden, wenngleich Kap. 7 einen Hinweis darauf liefert, daß sich Evolutionäre Algorithmen robust gegenüber lokaler Selektion verhalten. Auf jeden Fall kann aber der hier vorgestellte Modellierungsansatz als Brücke zwischen Theorie und Praxis dienen, um zu weiteren Erkenntnisse über nicht-panmiktische Evolutionäre Algorithmen zu gelangen und diese in Anwendung zu bringen.

Nur ein Formalismus? Ja, aber ein nützlicher.

Literatur

T. Bäck (1994). Selective pressure in evolutionary algorithms: a characterization of selection mechanisms. In Z. Michalewicz, J. D. Schaffer, H.-P. Schwefel, D. B. Fogel und H. Kitano (Hrsg.), *Proc. First IEEE Conf. Evolutionary Computation, IEEE World Congress on Computational Intelligence*, Band 1, Orlando FL, S. 57–62. IEEE Press, Piscataway NJ.

T. Bäck, D. B. Fogel und Z. Michalewicz (Hrsg.) (1997). *Handbook of Evolutionary Computation*. Oxford University Press, New York, und Institute of Physics Publishing, Bristol.

T. Bäck und M. Schütz (1995). Evolution strategies for mixed–integer optimization of optical multilayer systems. In J. R. McDonnell, R. G. Reynolds und D. B. Fogel (Hrsg.), *Evolutionary Programming IV – Proc. Fourth Annual Conf. Evolutionary Programming (EP'95)*, San Diego CA, S. 33–51. The MIT Press, Cambridge MA.

T. Bäck und M. Schütz (1996). Intelligent mutation rate control in canonical genetic algorithms. In Z. W. Raś und M. Michalewicz (Hrsg.), *Foundations of Intelligent Systems – Ninth Int'l Symp., ISMIS'96, Proc.*, Band 1079 der Reihe *Lecture Notes in Artificial Intelligence*, Zakopane, Polen, S. 158–167. Springer, Berlin.

J. E. Baker (1985). Adaptive selection methods for genetic algorithms. In J. J. Grefenstette (Hrsg.), *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, S. 101–111. Lawrence Erlbaum Associates, Hillsdale, NJ.

C. Berge (1973). *Graphs and hypergraphs.* (3. Auflage). North-Holland mathematical library ; 6. North-Holland Publ. Comp., Amsterdam.

H.-G. Beyer (1997). *Zur Analyse der Evolutionsstrategien*. Habilitationsschrift, Fachbereich Informatik, Universität Dortmund.

T. Blickle (1998). *Theory of Evolutionary Algorithms and Application to System Synthesis*. Dissertation, ETH Zürich.

G. Borgefors (1984a). Distance transformations in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing 27*, 321–345.

G. Borgefors (1984b). Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing 34*, 344–371.

E. Cantú-Paz und D. E. Goldberg (1997). Predicting speedups of ideal bounding cases of parallel genetic algorithms. In T. Bäck (Hrsg.), *Proc. 7th Int'l Conf. Genetic Algorithms*, San Francisco, S. 113–120. Morgan Kaufmann.

U. K. Chakraborty, K. Deb und M. Chakraborty (1996). Analysis of selection algorithms: A markov chain approach. *Evolutionary Computation 4*(2), 133–167.

J. P. Cohoon, S. U. Hedge, W. N. Martin und D. S. Richards (1987). Punctuated equilibria: A parallel genetic algorithm. In Grefenstette (1987), S. 148–154.

Y. Davidor (1991). A naturally occuring niche & species phenomenon: The model and first results. In R. K. Belew und L. B. Booker (Hrsg.), *Proceedings of the Fourth International Conference on Genetic Algorithms*, S. 257–263. Morgan Kaufmann Publishers, San Mateo, CA.

G. Dueck, T. Scheuer und H.-M. Wallmeier (1993). Toleranzschwelle und Sintflut: neue Ideen zur Optimierung. *Spektrum der Wissenschaft März 1993*, 42–51.

L. J. Eshelman (1997). Genetic algorithms. In T. Bäck, D. B. Fogel und Z. Michalewicz (Hrsg.), *Handbook of Evolutionary Computation*, S. B1.2:1–11. Oxford University Press, New York, und Institute of Physics Publishing, Bristol.

D. B. Fogel (1992). An analysis of evolutionary programming. In D. B. Fogel und W. Atmar (Hrsg.), *Proceedings of the First Annual Conference on Evolutionary Programming*, S. 43–51. Evolutionary Programming Society, San Diego CA.

L. J. Fogel, A. J. Owens und M. J. Walsh (1966). *Artificial Intelligence through Simulated Evolution*. Wiley, New York NY.

S. A. Furman und A. V. Tikhonravov (1992). *Basics of Optics of Multilayer Systems*. Basics of. Editions Frontières,Gif-sur-Yvette Cedex – France.

P. Gitchoff und G. P. Wagner (1996). Recombination induced hypergraphs: A new approach to mutation — recombination isomorphism. *Complexity 2*(1), 37—43.

D. E. Goldberg (1989). *Genetic algorithms in search, optimization and machine learning*. Addison Wesley, Reading, MA.

D. E. Goldberg und K. Deb (1991). A comparative analysis of selection schemes used in genetic algorithms. In G. J. E. Rawlins (Hrsg.), *Foundations of Genetic Algorithms*, S. 69–93. Morgan Kaufmann Publishers, San Mateo, CA.

V. Gordon (1994). Locality in genetic algorithms. In *Proc. First IEEE Int'l Conf. Evolutionary Computation (IEEE/ICEC'94), Band 1*, Orlando FA, S. 428–432. IEEE Press, Piscataway NJ.

M. Gorges-Schleuter (1989). ASPARAGOS an asynchronous parallel genetic optimization strategy. In J. D. Schaffer (Hrsg.), *Proc. Third Int'l Conf. Genetic Algorithms*, George Mason University, S. 422–427. Morgan Kaufmann.

M. Gorges-Schleuter (1990). *Genetic Algorithms and Population Structures — A Massively Parallel Algorithm.* Dissertation, Fachbereich Informatik, Universität Dortmund.

M. Gorges-Schleuter (1998). A comparative study of global and local selection in evolution strategies. In A. E. Eiben, T. Bäck, M. Schoenauer und H.-P. Schwefel (Hrsg.), *Parallel Problem Solving from Nature – PPSN V, Fifth Int'l Conf., Amsterdam, The Netherlands, September 27–30, 1998, Proc.*, S. 367–377. Springer, Berlin.

J. J. Grefenstette (Hrsg.) (1987). *Proc. Second Int'l Conf. Genetic Algorithms and Their Applications*. Lawrence Erlbaum Associates, Hillsdale, NJ.

T. P. Hartmann (1994). *Zellulare Hypergraphen und Ersetzungssysteme: Ein Konzept zur Modellierung dynamischer zellularer Strukturen*. Dissertation, Technische Hochschule Darmstadt.

F. Hoffmeister und T. Bäck (1990). Genetic algorithms and evolution strategies: Similarities and differences. interner Bericht "Grüne Reihe" No. 365, Fachbereich Informatik, Univ. Dortmund.

F. Hoffmeister und T. Bäck (1991). Genetic algorithms and evolution strategies: similarities and differences. In H.-P. Schwefel und R. Männer (Hrsg.), *Parallel Problem Solving from Nature – Proc. First Workshop PPSN*, Universität Dortmund, S. 455–469. Springer, Berlin, 1991.

F. Hoffmeister, G. Rudolph und J. Sprave (1991). DIOGENES: A fine–grain parallel algorithm with coarse–grain communication requirements. In *Papers and Posters presented at the Bonn Workshop*, Band Posters, Part P9, ESPRIT Parallel Computing Action, Universität Bonn. Project PCA 4117.

F. Hoffmeister und J. Sprave (1996). Problem–independent handling of constraints by use of metric penalty functions. In L. J. Fogel, P. J. Angeline und T. Bäck (Hrsg.), *Evolutionary Programming V – Proc. Fifth Annual Conf. Evolutionary Programming (EP'96)*, San Diego CA, S. 289–294. The MIT Press, Cambridge MA.

J. H. Holland (1975). *Adaptation in natural and artificial systems*. The University of Michigan Press, Ann Arbor MI.

A. Kapsalis, V. J. Rayward-Smith und G. D. Smith (1994). A unified paradigm for parallel genetic algorithms. In T. C. Fogarty (Hrsg.), *Evolutionary Computing*, Band 865 der Reihe *Lecture Notes in Computer Science*, Berlin, S. 131–149. Springer-Verlag.

A. J. Keane (1994). Experiences with optimizers in structural design. In I. C. Parmee (Hrsg.), *Proc. Conf. Adaptive Computing in Engineering Design and Control 94*, Plymouth, UK, S. 14–27.

S. Khuri, T. Bäck und J. Heitkötter (1994). The zero/one multiple knapsack problem and genetic algorithms. In E. Deaton, D. Oppenheim, J. Urban und H. Berghel (Hrsg.), *Proc. 1994 ACM Symp. Applied Computing*, Phoenix AZ, S. 188–193. ACM Press, New York.

J. Klockgether und H.-P. Schwefel (1970). Two–phase nozzle and hollow core jet experiments. In D. G. Elliott (Hrsg.), *Proc. Eleventh Symp. Engineering Aspects of Magnetohydrodynamics, 24.–26. März 1970*, California Institute of Technology, Pasadena CA, S. 141–148.

M. Laumanns, G. Rudolph und H.-P. Schwefel (1998). A spatial predator-prey approach to multi-objective optimization. In A. E. Eiben, T. Bäck, M. Schoenauer und H.-P. Schwefel (Hrsg.), *Parallel Problem Solving from Nature – PPSN V, Fifth Int'l Conf., Amsterdam, The Netherlands, September 27–30, 1998, Proc.*, S. 241–249. Springer, Berlin.

R. Lewandowski (1980). *Prognose- und Informationssysteme und ihre Anwendungen*, Band II. de Gruyter, Berlin - New York.

J. Mehnen (1994). *Parallel Evolution Strategies*. Diplomarbeit, Fachbereich Informatik, Universität Dortmund.

Z. Michalewicz (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, Berlin.

Z. Michalewicz, G. Nazhiyath und M. Michalewicz (1996). A note on usefulness of geometric crossover for numerical optimization problems. In L. J. Fogel,

P. J. Angeline und T. Bäck (Hrsg.), *Evolutionary Programming V – Proc. Fifth Annual Conf. Evolutionary Programming (EP'96)*, San Diego CA, S. 305–312. The MIT Press, Cambridge MA.

H. Mühlenbein und D. Schlierkamp-Voosen (1993). Predictive models for the breeder genetic algorithm. *Evolutionary Computation 1*(1), 25–49.

J. v. Neumann (1958). *The computer and the brain*. Yale University Press, New Haven und London.

J. K. Ousterhout (1994). *Tcl and the Tk Toolkit*. Addison-Wesley.

C. C. Pettey und M. R. Leuze (1989). A theoretical investigation of a parallel genetic algorithm. In Schaffer (1989), S. 398–405.

C. C. Pettey, M. R. Leuze und J. J. Grefenstette (1987). A parallel genetic algorithm. In Grefenstette (1987), S. 155–161.

M. Raith (1994). *Analyse von Routing-Blockaden mit Hilfe von Hypergraphen.* Dissertation, Univ., München.

L. A. Rastrigin (1974). *Sistemi ekstremalnowo iprawlenia*. Nauka, Moskau. (auf russisch).

I. Rechenberg (1965). Cybernetic solution path of an experimental problem. Royal Aircraft Establishment, Library translation No. 1122, Farnborough, Hants., UK.

I. Rechenberg (1973). *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann–Holzboog, Stuttgart.

S. Rolf (1998). Gemischt-ganzzahlige Optimierung mit Evolutionären Algorithmen am Beispiel des ARMA-Modells. Interner Bericht der Systems Analysis Research Group SYS–2/98, Universität Dortmund, Fachbereich Informatik.

H. H. Rosenbrock (1960). An automatic method for finding the greatest or least value of a function. *The Computer Journal 3*(3), 175–184.

G. Rudolph (1990). Globale Optimierung mit parallelen Evolutionsstrategien. Diplomarbeit, Universität Dortmund, Fachbereich Informatik.

G. Rudolph (1997). *Convergence Properties of Evolutionary Algorithms*. Verlag Dr. Kovač, Hamburg.

G. Rudolph und J. Sprave (1995). A cellular genetic algorithm with self–adjusting acceptance threshold. In *Proc. First IEE/IEEE Int'l Conf. Genetic Algorithms* *in Engineering Systems: Innovations and Applications (GALESIA'95)*, Sheffield, UK, S. 365–372. IEE Press, London.

G. Rudolph und J. Sprave (1996). Significance of locality and selection pressure in the grand deluge evolutionary algorithm. In H.-M. Voigt, W. Ebeling, I. Rechenberg und H.-P. Schwefel (Hrsg.), *Parallel Problem Solving from Nature – PPSN IV, Int'l Conf. Evolutionary Computation*, Berlin, S. 686–694. Springer, Berlin.

J. Sarma (1998). *An Analysis of Decentralized and Spatially Distributed Genetic Algorithms*. Dissertation, George Mason University, Fairfax, VA.

J. D. Schaffer (Hrsg.) (1989). *Proceedings of the Third International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers, San Mateo, CA.

L. Schönemann (1999). *Einfluß stochastischer Selektionsverfahren auf das Konvergenzverhalten von Evolutionsstrategien*. Diplomarbeit, Fachbereich Informatik, Universität Dortmund.

M. Schütz (1996). Eine Evolutionsstrategie für gemischt-ganzzahlige Optimierprobleme mit variabler Dimension. Interner Bericht der Systems Analysis Research Group SYS–1/96, Universität Dortmund, Fachbereich Informatik.

M. Schütz und J. Sprave (1996). Application of parallel mixed-integer evolution strategies with mutation rate pooling. In L. J. Fogel, P. J. Angeline und T. Bäck (Hrsg.), *Evolutionary Programming V – Proc. Fifth Annual Conf. Evolutionary Programming (EP'96)*, San Diego CA, S. 345–354. The MIT Press, Cambridge MA.

H.-P. Schwefel (1965). *Kybernetische Evolution als Strategie der experimentellen Forschung in der Strömungstechnik*. Diplomarbeit, Technische Universität Berlin, Hermann Föttinger–Institut für Strömungstechnik.

H.-P. Schwefel (1977). *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*, Band 26 der Reihe *Interdisciplinary Systems Research*. Birkhäuser, Basel.

H.-P. Schwefel (1995). *Evolution and Optimum Seeking*. Sixth-Generation Computer Technology. Wiley, New York, N.Y.

H.-P. Schwefel und F. Kursawe (1998). On natural life's tricks to survive and evolve. In *Proc. Second IEEE World Congress on Computational Intelligence (WCCI'98) with fifth IEEE Conf. Evolutionary Computation (IEEE/ICEC'98)*, Anchorage AK.

H.-P. Schwefel und G. Rudolph (1995). Contemporary evolution strategies. In F. Morán, A. Moreno, J. J. Merelo und P. Chacón (Hrsg.), *Advances in Artificial Life, Third European Conf. Artificial Life, Granada, Spain, June 1995, Proc.*, S. 893–907. Springer, Berlin.

H.-P. Schwefel, J. Sprave, M. Preuß, S. Rolf, R. Lewandowski und S. Theis (1997). Projekt `PARPROG` – Parallele Prognoseverfahren für die Absatzplanung. In R. Krahl und G. Wolf (Hrsg.), *Statustagung des BMBF:* HPSC'97 – Paralleles Höchstleistungsrechnen und seine Anwendungen, München, S. 211–223. Projektträger des BMBF für Informationstechnik bei der DLR e.V., Berlin.

U. Sedlag und E. Weinert (1987). *Woerterbuch der Biologie Biogeographie, Artbildung, Evolution*. VEB Gustav Fischer Verlag, Jena.

P. Spiessens und B. Manderick (1989). Fine-grained parallel genetic algorithms. In Schaffer (1989), S. 428–433.

J. Sprave (1990). *Parallelisierung Genetischer Algorithmen zur Suche und Optimierung*. Diplomarbeit, Universität Dortmund.

J. Sprave (1993). Zelluläre evolutionäre Algorithmen zur Parameteroptimierung. In R. Hofestädt, F. Krückeberg und T. Lengauer (Hrsg.), *Informatik in den Biowissenschaften, 1. Fachtagung der GI-FG 4.0.2*, Bonn, S. 111–120. Springer, Berlin.

J. Sprave (1994). Linear neighborhood evolution strategy. In A. V. Sebald und L. J. Fogel (Hrsg.), *Proc. Third Annual Conf. Evolutionary Programming (EP'94)*, San Diego CA, S. 42–51. World Scientific, Singapore.

J. Sprave (1999a). Ease: Evolutionary algorithms scripting environment. Reihe CI 54/99, Sonderforschungsbereich 531, Universität Dortmund.

J. Sprave (1999b). A unified model of non-panmictic population structures in evolutionary algorithms. In *Proc. Congress on Evolutionary Computation (CEC99)*, Washington DC. (im Druck).

J. Sprave und S. Rolf (1998). Variable-dimensional optimization with evolutionary algorithms using fixed-length representations. In V. W. Porto, N. Saravanan, D. Waagen und A. E. Eiben (Hrsg.), *Evolutionary Programming VII – Seventh Int'l Conf. EP'98, San Diego CA, March 25–27, 1998, Proc.*, S. 261–269. Springer, Berlin.

T. Starkweather, D. Whitley und K. Mathias (1991). Optimization using distributed genetic algorithms. In H.-P. Schwefel und R. Männer (Hrsg.), *Parallel*

Problem Solving from Nature — Proc. First Workshop PPSN I, Band 496 der Reihe *Lecture Notes in Computer Science*, S. 176–186. Springer, Berlin.

F. Stetter (1989). Modelling programs by the notion of hypergraph. *Software Engineering Journal* (3), 107 – 111.

R. Tanese (1987). Parallel genetic algorithm for a hypercube. In Grefenstette (1987), S. 177–183.

A. V. Thikonravov, M. K. Trubetskov, J. A. Dobrowolski und B. T. Sullivan (1995). Optimum solutions to single-band normal incidence antireflection coating problems. In *Optical Interference Coatings*, Band 17 der Reihe *OSA Technical Digest Series*, S. 49–51. Optical Society of America, Washington DC.

A. V. Tikhonravov und J. A. Dobrowolski (1993). Quasi-optimal synthesis for antireflection coatings: a new method. *Applied Optics 32*(22), 4265–4275.

D. Whitley (1989). The GENITOR algorithm and selection pressure: Why rank–based allocation of reproductive trials is best. In Schaffer (1989), S. 116–121.

R. R. Willey (1993). Predicting achievable design performance of broadband antireflection coatings. *Applied Optics 32*, 5447–5451.