

Das Räuber-Beute-Modell für die mehrkriterielle  
Optimierung – Analyse und Anwendung

**Dissertation**

zur Erlangung des Grades eines

DOKTORS DER INGENIEURWISSENSCHAFTEN

der Technischen Universität Dortmund  
an der Fakultät für Informatik

von  
Christian Grimme

Dortmund  
2012

**Tag der mündlichen Prüfung:**

02. Oktober 2012

**Dekanin:**

Prof. Dr. Gabriele Kern-Isberner

**Gutachter:**

Prof. Dr.-Ing. Uwe Schwiegelshohn, TU Dortmund

Prof. Dr.-Ing. Dr. Sc. h.c. Hans-Paul Schwefel, TU Dortmund

## Danksagung

Wie jedes auf längere Zeit angelegte Projekt, wäre diese Arbeit ohne die Unterstützung von vielen Seiten nicht möglich gewesen. Ich möchte mich deshalb bei all jenen bedanken, die mich auf dem langen Weg der Fertigstellung dieser Arbeit mit inspirierenden Vorschlägen, gutem Rat, konstruktiver Kritik und oftmals unendlicher Geduld begleitet haben.

Zuerst möchte ich Prof. Dr.-Ing. Uwe Schwiigelshohn danken, der mir am Institut für Roboterforschung – neben der alltäglichen Projektarbeit – über lange Jahre die nötige Freiheit zur Verfolgung meiner Forschungsinteressen gegeben hat. Als Doktorvater stand er mir zudem immer mit neuen Ideen, gutem Rat und fundierter Kritik zur Verfügung. Genauso dankbar bin ich Prof. Dr.-Ing. Dr. Sc. h.c. Hans-Paul Schwefel für die Übernahme der Begutachtung meiner Arbeit. Er war es auch, der mich während des Studiums mit dem Thema der evolutionären Optimierung konfrontierte und meine Arbeiten auch später mit großem Interesse verfolgte.

Arbeitskollegen sind besonders wichtig für eine gute und inspirierende Umgebung. In diesem Sinne möchte ich mich bei all meinen Kollegen am Institut für Roboterforschung für die gute Zusammenarbeit, viele gute fachliche Diskussionen, und die menschlich angenehme Arbeitsatmosphäre bedanken. Besonderer Dank gilt jedoch meinen engen Kollegen Joachim Lepping, Alexander Papaspyrou, Alexander Fölling und Markus Kemmerling. Die intensive und freundschaftliche Zusammenarbeit in dieser Gruppe hat sicherlich einen maßgeblichen Anteil an der Fertigstellung dieser Arbeit. Joachim Lepping möchte ich noch einmal besonders für seine inhaltliche Prüfung und fundierte Kommentierung der Arbeit danken.

Schließlich möchte ich meiner Familie und meinen Freunden für das Verständnis, die liebevolle Unterstützung und die fast unbegrenzte Geduld danken, die mir während der Fertigstellung dieser Arbeit entgegengebracht wurden. Insbesondere möchte ich Britta, Anita und Peter Grimme für die sorgfältigen sprachlichen Korrekturen sowie Felix Grimme für die intensive Prüfung vieler mathematischer Zusammenhänge der Arbeit danken.



Die vorliegende Arbeit ist auf der Grundlage der im Folgenden angegebenen Veröffentlichungen mit maßgeblicher Beteiligung des Autors entstanden. Einige der hier angegebenen Arbeiten befinden sich zum Zeitpunkt der Fertigstellung der Dissertation noch im Veröffentlichungsprozess. Insgesamt sind jedoch alle Arbeiten bereits begutachtet und für eine Veröffentlichung akzeptiert.

- GRIMME, C. & LEPPING, J. (2007). Designing Multi-Objective Variation Operators Using a Predator-Prey Approach. In *Proceedings of the Fourth International Conference on Evolutionary Multi-Criterion Optimization*, Band 4403 von *Lecture Notes in Computer Science*, (S. 21–35). Springer.
- GRIMME, C., LEPPING, J. & PAPASPYROU, A. (2007). Exploring the Behavior of Building Blocks for Multi-Objective Variation Operator Design using Predator Prey Dynamics. In Thierens, D. et al. (Hrsg.), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, (S. 805–812)., London. ACM Press.
- GRIMME, C., LEPPING, J. & PAPASPYROU, A. (2008). The Parallel Predator-Prey Model: A Step Towards Practical Application. In Rudolph, G. et al. (Hrsg.), *Proceedings of the 10th International Conference on Parallel Problem Solving From Nature (PPSN X)*, Band 5199 von *Lecture Notes in Computer Science*, (S. 681–690). Springer.
- GRIMME, C., LEPPING, J. & PAPASPYROU, A. (2009). Adapting to the Habitat: On the Integration of Local Search into the Predator Prey Model. In Ehrgott, M., Fonseca, C. M., Gandibleux, X., Hao, J.-K., & Sevaux, M. (Hrsg.), *Proceedings of the Fifth International Conference on Evolutionary Multi-Criterion Optimization (EMO)*, Band 5467 von *Lecture Notes in Computer Science*, (S. 510–524).
- GRIMME, C., LEPPING, J. & PAPASPYROU, A. (2011). Parallel Predator Prey Interaction for Evolutionary Multi-Objective Optimization. *Natural Computing*, 1–15 (online). online.
- GRIMME, C., LEPPING, J. & SCHWIEGELSHOHN, U. (2011). Multi-criteria scheduling: an agent-based approach for expert knowledge integration. *Journal of Scheduling*, 1–15 (online). online.
- GRIMME, C., KEMMERLING, M. & LEPPING, J. (2011). An expertise-guided multi-criteria Approach to Scheduling Problems. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, (S. 47–48). ACM Press.
- GRIMME, C. & LEPPING, J. (2011A). Combining basic heuristics for solving multi-objective scheduling problems. In *IEEE Symposium on Computational Intelligence in Scheduling (SCIS)*, (S. 9–16).

- GRIMME, C. & LEPPING, J. (2011B). Integrating Niching into the Predator Prey Model using epsilon-Constraints. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, (S. 109–110). ACM Press.
- GRIMME, C., KEMMERLING, M. & LEPPING, J. (2012). On the Integration of Theoretical Single-objective Scheduling Results for Multi-objective Problems. Buchkapitel in *EVOLVE – A Bridge Between Probability, Set Oriented Numerics, and Evolutionary Computation* (im Druck), Springer

## Abstract

The quest for improvement of systems is and has been an ubiquitous challenge for science as well as for technical application. Every engineer is confronted not only with the task of fulfilling a single optimality criterion but with the requirement to satisfy many often contradicting criteria as good as possible. He then strives for a set of so-called optimal compromises. Over the last 30 years, evolutionary algorithms have emerged as well-suited heuristics to solve this task. Nowadays, this area provides a set of standard approaches which usually yield good approximation results. However for engineers without detailed insight into an algorithm's structure, most of the monolithic approaches are complex to apply.

In this work, we thus focus on a simple and also nature-inspired alternative approach: the Predator Prey Model for multi-objective optimization. Unlike the established algorithms, this approach comprises local acting single-objective components, which make it rather modular. That property enables parallelization and an application-specific configuration. Surprisingly, the dynamics of the approach are not well understood, such that a conclusion regarding its application potential is still open. Therefore, a first task of this work is the observation of internal processes and their theoretical description. With that background at hand, we propose and evaluate two exemplary extensions of the model and show their benefit for approximation quality. Finally, we transfer the model to the area of multi-objective scheduling and show its superiority to established models. In this context we demonstrate, how the modular character of the Predator-Prey Model can be used to establish a framework for the integration of heuristic expert knowledge.



## Zusammenfassung

Das Streben nach Verbesserung von Systemen ist schon immer eine allgegenwärtige Herausforderung in Wissenschaft und technischer Anwendung. In der Regel ist jeder Ingenieur nicht nur mit dem Problem konfrontiert, ein einziges Ziel möglichst gut zu erfüllen; für die Entwicklung realer Systeme sind oftmals mehrere sich widersprechende Ziele (oder Kriterien) zugleich zu erfüllen. So ergibt sich für die mehrkriterielle Optimierung die Aufgabe, eine Menge sogenannter optimaler Kompromisse aufzufinden, die alle Kriterien zugleich möglichst gut erfüllen. In den letzten 30 Jahren haben evolutionäre Algorithmen als gute Heuristiken für diesen Bereich an Bedeutung gewonnen. Dort sind Standardverfahren entwickelt worden, die gute Approximationen der Kompromissmenge liefern, jedoch oftmals monolithisch aufgebaut sind und eine für die Anwenderschaft hohe Komplexität aufweisen.

In dieser Arbeit betrachten wir eine einfache und ebenfalls naturinspirierte Alternative zu den etablierten Verfahren: das Räuber-Beute-Modell für die mehrkriterielle Optimierung. Anders als die Standardansätze basiert es auf lokal agierenden einkriteriellen algorithmischen Komponenten und ist sehr modular. Dies erleichtert neben der Parallelisierung des Ansatzes auch die flexible und anwendungsbezogene Konfiguration. Zugleich ist es jedoch ein Verfahren, dessen Dynamik noch weitgehend unverstanden ist, so dass eine Abschätzung über das Potential des Ansatzes (über die zuvor genannte Flexibilität und Einfachheit in der Anwendung hinaus) bisher aussteht. Die erste Aufgabe in dieser Arbeit ist es daher, die inneren Prozesse des Räuber-Beute-Modells zu beobachten und theoretisch zu beschreiben. Erst mit diesem Wissen stellen wir exemplarisch zwei Erweiterungsmöglichkeiten des Verfahrens vor und zeigen deren Nutzen. Schließlich übertragen wir das Modell in die Anwendung für den Bereich des mehrkriteriellen Scheduling und stellen dar, dass es nicht nur etablierten Verfahren überlegen ist, sondern durch seinen modularen Aufbau auch als Framework zur Integration von heuristischem Expertenwissen gewinnbringend angewendet werden kann.



# Inhaltsverzeichnis

<b>Einleitung</b>	<b>xiii</b>
<b>I Optimierung und evolutionäre Algorithmen: Grundlagen</b>	<b>1</b>
<b>1 Grundlagen der Optimierung</b>	<b>3</b>
1.1 Definition von Optimierungsproblemen . . . . .	5
1.2 Prinzipien von Optimierungsmethoden . . . . .	8
1.2.1 Analytischer Ansatz . . . . .	9
1.2.2 Numerischer Ansatz . . . . .	11
<b>2 Evolutionäre Verfahren</b>	<b>15</b>
2.1 Grundlagen und Prinzipien . . . . .	16
2.2 Notation und Evolutionsschleife . . . . .	17
2.2.1 Repräsentationen von Individuen . . . . .	18
2.2.2 Variationsoperatoren . . . . .	20
2.2.3 Die Rolle der Selektion . . . . .	25
2.2.4 Populationsstrukturen . . . . .	28
<b>3 Mehrkriterielle Algorithmen</b>	<b>33</b>
3.1 Deterministische Ansätze . . . . .	34
3.1.1 Methode der Gewichtung . . . . .	34
3.1.2 Epsilon-Constraint-Methode . . . . .	34
3.1.3 Tchebycheff-, Zenitpunkt- oder Compromise Programming-Methode . . . . .	35
3.2 Mehrkriterielle Evolutionäre Verfahren . . . . .	37
3.2.1 Dominanzbasierte Methoden . . . . .	38
3.2.2 Indikatorbasierte Methoden . . . . .	44
3.2.3 Alternative Methoden . . . . .	46

<b>II</b>	<b>Das Räuber-Beute-Modell für die mehrkriterielle Optimierung und die Untersuchung seiner Eigenschaften</b>	<b>49</b>
<b>4</b>	<b>Die Idee des Räuber-Beute-Modells</b>	<b>51</b>
4.1	Schwefels Räuber-Beute-Modell . . . . .	53
4.2	Varianten und verwandte Ansätze . . . . .	56
4.3	Ein formales und modifiziertes Modell . . . . .	59
<b>5</b>	<b>Analyse des Räuber-Beute-Modells</b>	<b>63</b>
5.1	Modularität und Parallelisierbarkeit . . . . .	63
5.2	Räuber-Beute-Interaktion . . . . .	68
5.2.1	Einige Vorbereitungen . . . . .	68
5.2.2	Analyse des Mutationseinflusses . . . . .	70
5.2.3	Analyse des Rekombinationseinflusses . . . . .	80
5.2.4	Kombination von Mutation und Rekombination . . . . .	86
<b>III</b>	<b>Erweiterungen des Räuber-Beute-Modells</b>	<b>91</b>
<b>6</b>	<b>Grenzen – Vorstudie und Kritik</b>	<b>93</b>
6.1	Testprobleme und Versuchsaufbau . . . . .	93
6.2	Ergebnisse . . . . .	95
6.3	Diskussion und kritische Betrachtung . . . . .	96
6.3.1	Modulare Lösungsbildung im Suchraum . . . . .	97
6.3.2	Die Problematik nicht-konvexer Probleme . . . . .	98
<b>7</b>	<b>Integration Gradienten-basierter Suche</b>	<b>101</b>
7.1	Approximation der Gradienteninformation . . . . .	102
7.2	Bestimmung des mehrkriteriellen Abstiegs . . . . .	104
7.3	Hybridisierung des Modells über die Beute . . . . .	106
7.4	Evaluation des Hybridansatzes . . . . .	108
7.4.1	Ausschließliche Betrachtung der lokalen Suche . . . . .	108
7.4.2	Anwendung auf ein nicht-konvexes Problem . . . . .	111
7.5	Diskussion der Erweiterung . . . . .	116
<b>8</b>	<b>Diversitätserhaltung durch <math>\varepsilon</math>-Constraints</b>	<b>119</b>
8.1	Integration von Zielbeschränkungen . . . . .	119
8.2	Erweiterungen und Implementierung . . . . .	122
8.2.1	Automatisches Auffinden des Zielgebiets im Entscheidungsraum . . . . .	123
8.2.2	Nischenbildung durch Räubereinfluss . . . . .	125
8.3	Evaluation des Ansatzes . . . . .	126
8.4	Diskussion der Erweiterung . . . . .	129

<b>IV</b>	<b>Anwendung in der diskreten Optimierung - Eine Fallstudie im mehrkriteriellen Scheduling</b>	<b>131</b>
<b>9</b>	<b>Mehrkriterielles Scheduling</b>	<b>133</b>
9.1	Scheduling und Notation . . . . .	134
9.2	Einkriterielle Schedulingprobleme . . . . .	135
9.3	Mehrkriterielle Schedulingprobleme . . . . .	137
<b>10</b>	<b>Ein- und Mehrmaschinenprobleme</b>	<b>139</b>
10.1	Transfer des Räuber-Beute-Modells . . . . .	140
10.1.1	Repräsentation von Schedulingproblemen . . . . .	140
10.1.2	Ein Variationsoperator für Schedulinganwendungen . . . . .	141
10.1.3	Die Kopplung von Räubern und Variationsoperatoren . . . . .	142
10.2	Anwendung auf ein Einmaschinenproblem . . . . .	144
10.3	Anwendung auf Mehrmaschinenprobleme . . . . .	146
10.3.1	Probleminstanzen, ein Vergleichsalgorithmus und Auswertungsvorgehen . . . . .	146
10.3.2	Das Problem $P_k    C_{max}, \sum C_j$ . . . . .	148
10.3.3	Das Problem $P_k    \sum C_j, \sum U_j$ . . . . .	150
10.4	Modularität und Erweiterbarkeit . . . . .	152
10.4.1	Erhöhung der Problemkomplexität . . . . .	152
10.4.2	Erhöhung der Kriterienanzahl . . . . .	154
10.5	Zusammenfassung und Diskussion . . . . .	156
	<b>Zusammenfassung der Arbeit und Ausblick</b>	<b>159</b>
	<b>Anhang</b>	<b>165</b>
A.1	Zum Auffinden nicht-konvexer Lösungen durch die Gewichtungsmethode . . . . .	165
A.2	Die ZDT-Testsuite . . . . .	166
A.3	Evaluation Gradientensuche . . . . .	167
A.4	Referenzergebnisse etablierter Methoden für das Kursawe-Problem . . . . .	167
A.5	Konfigurationen für den $\epsilon$ -Constraint-Ansatz . . . . .	168
A.6	Schedulingproblem, Instanz und Lösungen . . . . .	169



# Einleitung

Jede Entscheidung im privaten, geschäftlichen oder politischen Umfeld soll so getroffen werden, dass möglichst alle verfolgten Ziele gleichzeitig und so gut wie eben möglich erfüllt sind. Jedoch führt der Entscheidungsprozess immer in einen Gleichgewichtszustand, der sich abstrakt folgendermaßen beschreiben lässt:

Ein Gleichgewichtszustand zwischen verschiedenen Zielen ist erreicht, wenn sich *alle* verfolgten Ziele *zugleich* nicht besser erfüllen lassen als in dem Gleichgewichtszustand.

Mag diese Definition zuerst verwirrend erscheinen, so ist sie doch einleuchtend, wenn man die alltägliche Entscheidungsproblematik betrachtet. Etwa der Kauf eines Produktes entscheidet sich oftmals nach mehreren Kriterien, die durchaus gegensätzlicher Natur sein können: ein einfaches Beispiel sind hier ein hoher Qualitätsanspruch an und eine geringe Preisvorstellung für das Produkt. Allein die Erfahrung lehrt uns aber: In einem Gleichgewichtszustand muss immer ein Ziel für eine Verbesserung des anderen Zieles teilweise aufgegeben werden. Auch Politiker befinden sich in der oft aussichtslos erscheinenden Lage nicht alle Bevölkerungsgruppen zugleich durch geeignete Entscheidungen bedienen und in eine bessere Lebenssituation versetzen zu können. Eine Veränderung hin zu einer Verbesserung für die eine Gruppe führt mitunter zwangsläufig zu einer Verschlechterung für eine andere Gruppe. Die Gleichgewichtszustände sind immer Kompromisse, die sich im Wechselspiel politischer Kräfte verschieben, jedoch niemals alle Ziele gleichermaßen gut erfüllen können.

Wie diese Betrachtung gezeigt hat, gibt es oft viele solcher Kompromisse, die einander ersetzen können jedoch niemals im Sinne des Gleichgewichts besser sind als andere. Um trotz dieser Unvergleichbarkeit der Kompromisse dennoch eine fundierte Entscheidung – vielleicht nur aufgrund individueller Vorstellungen oder des Bauchgefühls – treffen zu können, ist es wünschenswert im Idealfall alle Kompromisse zu kennen. Ist ein formaler Optimierungsprozess im alltäglichen Leben und auch in der Politik eher nebensächlich, so spielt eine algorithmische Herangehensweise in der Entwicklung technischer Systeme oder in der Planung von regelbasierten Abläufen eine große Rolle. Daher ergibt sich für die Algorithmik die Herausforderung, alle Gleichgewichtszustände oder „optimalen Kompromisse“ zur Verfügung zu stellen.

Seit der 1905 entstandenen formalen Begründung der mehrkriteriellen Optimierung durch Paretos Diskussion des Themas im ökonomisch-politischen Kontext (Pareto, 1971, engl. Übersetzung) wurde eine Vielzahl an algorithmischen Ansätzen vorgeschlagen, die dazu dienen sollen, die als Paretomenge bezeichnete Menge der optimalen Kompromisse zu bestimmen. Neben einer theoretisch fortgeschrittenen Behandlung – die zugleich Grundlage für viele Lösungsansätze ist – sind einige generelle deterministische Verfahren entwickelt worden (Miettinen, 1999), die sich auf eine große Menge von Problemstellungen anwenden lassen. Aber auch randomisierte Verfahren, speziell der Bereich der mehrkriteriellen evolutionären Algorithmen, spielt bei der Lösung von schweren Problemen eine zunehmend wichtige Rolle. Seit den 1980er Jahren werden die Prinzipien der evolutionären Optimierung erfolgreich auch auf den Bereich der Mehrzieloptimierung übertragen (Deb, 2001; Coello, van Veldhuizen & Lamont, 2007). Dies bietet sich grundsätzlich an, da evolutionäre Algorithmen auf sogenannten Populationen von Lösungen basieren und so in einem Schritt mehrere Kompromisse betrachten oder erzeugen können. Es muss also „lediglich“ sichergestellt werden, dass diese Lösungen optimale Kompromisse darstellen. Leider ist es im Allgemeinen nicht einfach die gesamte Lösungsmenge, also eine einerseits exakte und zugleich vollständige optimalen Kompromissmenge zu erzeugen. Sicherlich sind auch deshalb viele verschiedene Ansätze entstanden, die teilweise sehr ausgefeilte Mechanismen einsetzen, um beide Ziele, Vollständigkeit und Genauigkeit, zu erreichen. Grundlegend für alle Ansätze ist das Fundament der Evolutionsschleife aus Reproduktion, Variation und Selektion, wobei insbesondere in der Selektion sehr verschiedene und oftmals komplexe Ansätze verfolgt werden, um mit dem Problem der Unvergleichbarkeit von Lösungen umgehen zu können. Wurden zu Beginn der Forschung im Bereich der evolutionären mehrkriteriellen Optimierung noch einzelne Kriterien betrachtet, so werden inzwischen vorwiegend Dominanz-basierte oder Indikator-geleitete Methoden eingesetzt<sup>1</sup>.

Trotzdem sind neben dieser Hauptlinie der Weiterentwicklung algorithmischer Methoden andere Ideen entstanden, die die Einfachheit und Übertragbarkeit des algorithmischen Konzeptes in die praktische Anwendung anstreben. Ein Beispiel eines solchen algorithmisch einfachen Ansatzes ist die Idee von Laumanns, Rudolph & Schwefel (1998), die Interaktion von Räuber und Beute aus der Natur zu entlehnen. In diesem Ansatz stellen die Räuber einzelne Kriterien dar, die alle gemeinsam auf eine Population von Beuteindividuen wirken. Dadurch wird von allen Kriterien ein Anpassungsdruck auf die Beuteindividuen ausgeübt, der diese – so zumindest die Vorstellung – zu einer Adaptation gegenüber aller Kriterien zugleich, also zu möglichst guten Kompromissen führt. Neben der offensichtlichen Einfachheit der grundlegenden Methodik, bietet der Ansatz ein großes Potential für die parallele Ausführung. Gerade mit dem Aufkommen von Mehrkernprozessoren und dem zunehmenden Einsatz vernetzter Großrechner gewinnt der modulare

---

<sup>1</sup>Eine genauere Darstellung der Methoden erfolgt in Abschnitt 3.2.

und entkoppelte Räuber-Beute-Ansatz auch an praktischer Relevanz gegenüber etablierten oft monolithischen Methoden.

Umso erstaunlicher ist, dass es zu diesem Ansatz bisher nur eine sehr überschaubare Anzahl von Arbeiten gibt. Obwohl das Räuber-Beute-Modell bereits vor mehr als fünfzehn Jahren von Schwefel in seinen Lehrveranstaltungen vorgeschlagen wurde und inzwischen einige Erweiterungen existieren, haben sich andere Autoren bisher überraschend wenig mit der zugrunde liegenden Dynamik, dem Zusammenwirken der beteiligten Komponenten und der Anwendbarkeit des Algorithmus beschäftigt. Zwar wurden die einfache Realisierbarkeit und auch die Entkopplung der Komponenten selbst hervorgehoben, der Nutzen dieser Eigenschaften wurde jedoch nicht betrachtet.

## Zielsetzung

In der vorliegenden Arbeit wird das sogenannte Räuber-Beute-Modell von Laumanns, Rudolph und Schwefel folgerichtig in zweierlei Hinsicht betrachtet.

Einerseits wird versucht ein genaueres Verständnis der im Algorithmus vorherrschenden Dynamik und ihrer Auswirkung auf die Lösungsbildung zu erlangen. Dies ist eine ganz wichtige Grundlage, um die Verwendbarkeit des Ansatzes überhaupt beurteilen zu können. Zudem kann nur ein tieferes Verständnis der Eigenschaften und innewohnenden Prozesse dazu führen, sinnvolle und begründete Erweiterungen des Modells vorzuschlagen.

Andererseits streben wir an, die Vorteile der Modularisierbarkeit des Modells auszunutzen und diese in einer praxisnahen Anwendung zu evaluieren. Dabei verwenden wir exemplarisch mehrkriterielle Schedulingprobleme und zeigen, wie ein-kriterielles Wissen leicht und gewinnbringend in die algorithmische Struktur integriert werden kann.

## Struktur und Methodik der Arbeit

Die Arbeit gliedert sich in vier Teile. Der erste Teil schafft die Grundlage für die Einordnung des Ansatzes in das Gebiet der Optimierung und der evolutionären mehrkriteriellen Ansätze. Der zweite Teil führt die Idee des Räuber-Beute-Modells ausführlich ein und diskutiert bisherige Erweiterungsansätze anderer Autoren. Daraufhin wird ein formales und zudem gegenüber Laumanns' Ansatz leicht modifiziertes Modell zur Betrachtung in dieser Arbeit festgelegt. Davon ausgehend diskutieren wir die Eigenschaften der Modularität und Parallelisierbarkeit, bevor wir uns einer ausführlichen Analyse der Modelldynamik entlang des Einflusses des Variationsoperatoren Mutation und Rekombination auf reellwertigen Problemen zuwenden. Diese Ergebnisse ermöglichen es uns im dritten Teil der Arbeit eine erste Bewertung des Modells vorzunehmen sowie anschließend zwei Erweiterungsansätze – basierend auf lokaler gradientenbasierter Suche und auf der Einbeziehung von Zielbeschränkungen ( $\varepsilon$ -Constraints) – vorzuschlagen. Der vierte und letzte Teil

der Arbeit wendet das Räuber-Beute-Modell schließlich auf Probleme des mehrkriteriellen Scheduling an. Dort wird das zuvor erlangte Verständnis über die Dynamik und die Einsicht in die Eigenschaft der Modularität genutzt, um exemplarisch Ein- und Mehrmaschinenprobleme durch die Integration einkriteriellen Expertenwissens zu lösen.

Teil I

# Optimierung und evolutionäre Algorithmen: Grundlagen



# Kapitel 1

## Grundlagen der Optimierung

Das Bemühen um Weiterentwicklung und Verbesserung von Werkzeugen, Prozessen, Organisationsformen, sogar vollständigen Systemen spielt bereits seit tausenden von Jahren eine wichtige Rolle im menschlichen Dasein. So wurden – verstärkt durch methodische und wissenschaftliche Betrachtung – vielfältige Verfahren entwickelt, die die Problematik der Optimierung in strukturierter und gut begründeter Weise angehen. Abstrahiert man von speziellen Aspekten und Lösungen einzelner Wissenschaftsdisziplinen, so kann der Vorgang der Optimierung dadurch beschrieben werden, dass versucht wird aus einer Anzahl von Möglichkeiten die beste zu wählen. Diese beste Alternative wird als *Optimum* bezeichnet und als *Lösung* einer Optimierung angestrebt. Wichtige Elemente der Optimierung und Voraussetzung für die Entwicklung von Optimierungsmethoden sind veränderbare und ein System beschreibende *Parameter*, ein als *Zielfunktion* bezeichneter funktionaler Zusammenhang zwischen Parametern und Systemverhalten sowie eine geeignete *Strategie* zur qualitativen Unterscheidung und Auswahl von Alternativen.

Trotz dieser abstrakten Beschreibung eines jeden Optimierungsproblems ist eine universelle Methode zur Behandlung aller Problemstellungen bisher nicht entwickelt worden. Vielmehr haben sich in der mathematischen und informationstechnischen Betrachtung Klassifizierungen herausgebildet, die zugleich Voraussetzungen und methodische Ansatzpunkte für verschiedene Optimierungsverfahren beschreiben. So unterscheidet man zwischen experimenteller und mathematischer Optimierung, aber auch zwischen parameterbezogener und funktionaler, statischer und dynamischer sowie numerischer und analytischer Optimierung (Schwefel, 1995).

Die Unterscheidung experimenteller und mathematischer Optimierung bezieht sich insbesondere auf die dem Problem zugrunde liegende Beschreibbarkeit des betrachteten Systems. Liegen keine funktionalen Informationen über Beziehungen zwischen Parametern und Zielen vor, so ist eine systematische Betrachtung nur am tatsächlichen zu optimierenden System selbst oder einem sehr realitätsnahen, nicht mathematischen Modell möglich (Black-Box-Optimierung). Ist jedoch der Bezug zwischen Zielfunktion und Parametern gegeben, so ist die direkte Manipulation

von Parametern und Zielfunktionen möglich. In diesem Kontext ist die Klassifizierung parameterbezogener und funktionaler Optimierung anzutreffen. Sind Parameter und Zielfunktionen skalare Werte, so spricht man von Parameter-bezogener Optimierung. Werden die Parameter hingegen selbst als Funktionen ausgedrückt, handelt es sich um Probleme der Funktionenoptimierung.<sup>1</sup>

Ein anderer Aspekt der Klassifizierung von Optimierungsproblemen beleuchtet die zeitliche Dynamik eines zu ermittelnden Optimums. Bleibt dieses im Laufe der Zeit unverändert, spricht man von statischer Optimierung. Verändert es aber seine Lage abhängig von Umgebungsbedingungen des Systems während des Optimierungsprozesses, so wird dieser Typ von Problemen als dynamisch bezeichnet.

Die noch verbleibende Klassifizierung von analytischer und numerischer Optimierung wird – etwas ungewöhnlich – nicht direkt über Problemeigenschaften, sondern vielmehr über die verwendeten Methoden zur Lösungsbildung vorgenommen. Zugleich stellt sie einen idealen Übergang zur Beschreibung der in der Arbeit betrachteten Verfahren dar: Abhängig vom Suchvorgehen bei der Bestimmung eines Optimums wird eine Methode als numerisch oder analytisch bezeichnet. Numerische Verfahren versuchen ausgehend von einer aktuellen Lösung einen den Zielfunktionswert verbessernden Schritt durchzuführen und damit der optimalen Lösung näher zu kommen. Bei der Entscheidung für eine Suchrichtung wird dabei nicht notwendigerweise auf Eigenschaften der Zielfunktion zurückgegriffen. Es werden oftmals die Parameter variiert und anschließend auf Grundlage der Zielfunktionsauswertung entschieden, ob eine Suchrichtung weiter verfolgt wird. Analytische Verfahren hingegen versuchen die analytischen Eigenschaften der Zielfunktion an der Stelle eines Optimums auszunutzen. Eine einfache Eigenschaft differenzierbarer Zielfunktion ist etwa die notwendige Bedingung des Verschwindens der ersten partiellen Ableitungen in einem lokalen oder globalen Optimum. Sie kann ausgenutzt werden, um mögliche Kandidaten für eine optimale Lösung zu bestimmen. Für das Auffinden der Kandidaten ist es anders als bei den numerischen Verfahren nicht nötig einen Test der Güte durchzuführen. Erst die Kandidaten selbst können mit einem hinreichenden Kriterium (im Fall des Beispiels die Überprüfung der zweiten partiellen Ableitung) auf Optimalität überprüft werden, siehe auch Abschnitt 1.2.1. Im Allgemeinen sind jedoch auch die analytischen Verfahren auf iterative Methoden oder sogar die schrittweise Betrachtung der Eigenschaften der Zielfunktion angewiesen.<sup>2</sup>

---

<sup>1</sup>Ziel dieser Art der Optimierung ist es zumeist nicht nur einzelne Punkte, sondern ganze Trajektorien im Lösungsraum zu finden.

<sup>2</sup>Bei Zielfunktionen zweiter Ordnung kann bereits die Lösung der entstehenden linearen Gleichungssysteme nicht mehr in einem Schritt durchgeführt werden. Für Zielfunktionen höherer Ordnung setzt die Betrachtung der notwendigen Bedingung bereits die Lösung nicht-linearer Gleichungen voraus. Die Bestimmung der Nullstellen potentieller Extremalösungen kann dann nur noch iterativ über wiederholte Relaxation mittels Interpolation oder anderen Approximationsmethoden geschehen.

## 1.1 Definition von Optimierungsproblemen

Entsprechend der vorherigen allgemeinen Beschreibung und unabhängig von den unterschiedlichen Klassifizierungen ergibt sich ein Optimierungsproblem bezüglich eines einzelnen Kriteriums auf einem Suchraum  $S$  und einem Zielraum  $Z$  über:

- $f : S \rightarrow Z$
- minimiere  $f$  (auf  $S$ )

Dabei beschreibt  $f$  als Optimierungsproblem die funktionale Abhängigkeit zwischen Elementen  $\vec{x} \in S \subseteq \mathbb{R}^n$  aus dem Entscheidungsraum und Qualitätswerten aus dem Zielraum  $Z \subseteq \mathbb{R}$ . Die Zahl  $n$  wird Dimension des Problems genannt. An dieser Stelle ist noch nicht angegeben, ob die funktionalen Abhängigkeiten  $f$  für ein Problem tatsächlich bekannt sind oder nicht.<sup>3</sup> Es ist also noch keine Festlegung getroffen worden, ob es sich um ein Problem der experimentellen oder mathematischen Optimierung handelt. Andererseits ist das Problem bereits als Minimierungsproblem definiert. Dies ist ohne Einschränkungen aufgrund des Dualitätsprinzips von Optimierungsproblemen möglich, siehe Gleichung 1.1. Daher werden hier nur Minimierungsprobleme betrachtet.

$$\max f = - \min (-f) \tag{1.1}$$

In einigen Fällen wird ein Optimierungsproblem so beschrieben, dass der Entscheidungsraum  $S$  durch Randbedingungen  $g_i$ ,  $i = 1, \dots, k$  auf  $G := \{\vec{x} \in \mathbb{R} \mid g_i(\vec{x}) \leq 0 \forall i = 1, \dots, k\}$  eingeschränkt wird. Wir suchen also in diesem Falle nicht mehr auf  $S$  sondern auf  $S \cap G$ .

Für das gerade eingeführte Optimierungsproblem mit einem einzigen Ziel ergibt sich eine natürliche Bewertungsmethode der Lösungen aufgrund der Ordnung von  $Z \subseteq \mathbb{R}$ . Im Falle eines mehrkriteriellen Problems mit  $m$  Zielen ist der Zielraum eine Teilmenge von  $\mathbb{R}^m$ . Es muss also eine weitergehende Ordnungsrelation definiert werden, um schließlich zu einem allgemeineren Optimalitätsbegriff zu gelangen und damit das mehrkriterielle Optimierungsproblem selbst definieren zu können.

### Definition 1.1 (Mehrkriterielle Dominanz)

Sei  $f : S \rightarrow Z$ ,  $S \subseteq \mathbb{R}^n$ ,  $Z \subseteq \mathbb{R}^m$  und  $\vec{x}, \vec{y} \in S$ .

1.  $\vec{x}$  dominiert  $\vec{y}$  (bzgl.  $f$ )  $:\Leftrightarrow f_i(\vec{x}) \leq f_i(\vec{y}) \forall i = 1, \dots, m$ .

Notation:  $\vec{x} \preceq \vec{y}$

2.  $\vec{x}$  dominiert  $\vec{y}$  echt (bzgl.  $f$ )  $:\Leftrightarrow \vec{x} \preceq \vec{y}$  und  $f_i(\vec{x}) < f_i(\vec{y})$  für ein  $1 \leq i \leq m$ .

Notation:  $\vec{x} \prec \vec{y}$

---

<sup>3</sup>Zwar haben wir  $f$  abstrakt als Beschreibung der funktionalen Abhängigkeit eingeführt. In der Realität ist eine einfache mathematische Beschreibung von  $f$  jedoch oft nicht gegeben.

Im Gegensatz zur einkriteriellen Dominanz ( $\leq$ ) beschreibt die mehrkriterielle Dominanz keine Ordnung. Die Relation  $\preceq$  beschreibt eine partielle Ordnung<sup>4</sup> auf  $S$ , während die Relation  $\prec$  nicht reflexiv und damit keine partielle Ordnung ist. Diese Feststellung macht es nötig, den gewöhnlicherweise auf Skalarität beruhenden Optimalitätsbegriff in einer anderen Art zu fassen. Pareto (1971) hat diesen neuen Begriff eingeführt und bezeichnet eine Lösung als optimal, wenn eine Verbesserung eines Teilziels der Lösung nur durch die Verschlechterung mindestens eines anderen Teilziels erreicht wird. Formal wird die sogenannte Pareto-Optimalität in der folgenden Definition ausgedrückt.

**Definition 1.2 (Pareto-Optimalität)**

Unter den Annahmen in Definition 1.1 heißt  $\vec{x}^* \in S$  Pareto-optimal bezüglich  $f$ , wenn  $\vec{x}^*$  von keinem Element aus  $S$  echt dominiert wird. Die Menge  $\mathcal{X}^*$  aller Pareto-optimalen Elemente aus  $S$  wird Paretomenge bezüglich  $f$  genannt und formal folgendermaßen ausgedrückt:

$$\mathcal{X}^* := \mathcal{X}_f^* := \{\vec{x} \in S \mid \nexists \vec{y} \in S : \vec{y} \prec \vec{x}\}.$$

Die Menge  $f(\mathcal{X}_f^*)$ , das Bild der Paretomenge unter  $f$ , bezeichnen wir als Pareto-front.

Der Vollständigkeit halber sei hier noch ein etwas weiterer Begriff der Pareto-Optimalität angeführt. Die Menge der schwach Pareto-optimalen Lösungen enthalte all jene Elemente des Entscheidungsraums (Entscheidungsvektoren), für die kein weiterer Vektor zu finden ist, der bezüglich aller Zielkriterien besser ist. Durch die damit verbundene Aufweichung des Dominanzkriteriums<sup>5</sup> ergibt sich eine Obermenge der Pareto-optimalen Lösungen, die manchmal leichter zu ermitteln ist als die Paretomenge selbst. Gehen wir wieder von den in Definition 1.1 getroffenen Annahmen aus, so ist dieser Begriff wie im Folgenden dargestellt definiert. Siehe auch Abbildung 1.1 für ein Beispiel.<sup>6</sup>

**Definition 1.3 (Schwache Pareto-Optimalität)**

Ein Entscheidungsvektor  $\vec{x}^{*,w} \in S$  heißt schwach Pareto-optimal bezüglich  $f$ , wenn es keinen Entscheidungsvektor  $\vec{y} \in S$  gibt mit  $f_i(\vec{y}) < f_i(\vec{x}^{*,w})$  für alle  $i = 1, \dots, m$ . Gemäß der Notation in Definition 1.2 schreiben wir:

$$\mathcal{X}^{*,w} := \mathcal{X}_f^{*,w} := \{\vec{x} \in S \mid \nexists \vec{y} \in S : f_i(\vec{y}) < f_i(\vec{x}) \forall i = 1, \dots, m\}.$$

Betrachtet man die lokalen Eigenschaften eines mehrkriteriellen Optimierungsproblems, so muss man auch das Konzept der lokalen Pareto-Optimalität betrachten.

<sup>4</sup>Da in der Literatur keine einheitlichen Begrifflichkeiten verwendet werden, ist hier eine transitive und reflexive Relation gemeint.

<sup>5</sup>In einigen Fällen wird daher auch von schwacher Dominanz gesprochen.

<sup>6</sup>Natürlich ist die Paretofront nicht, wie dargestellt, grundsätzlich kontinuierlicher Art. Bei kombinatorischen Problemen besteht sie immer aus diskreten Lösungen. Die gewählte Darstellung ist hier zur Veranschaulichung vereinfacht. Tatsächliche Darstellungen von Paretofronten finden sich im weiteren Verlauf der Arbeit.

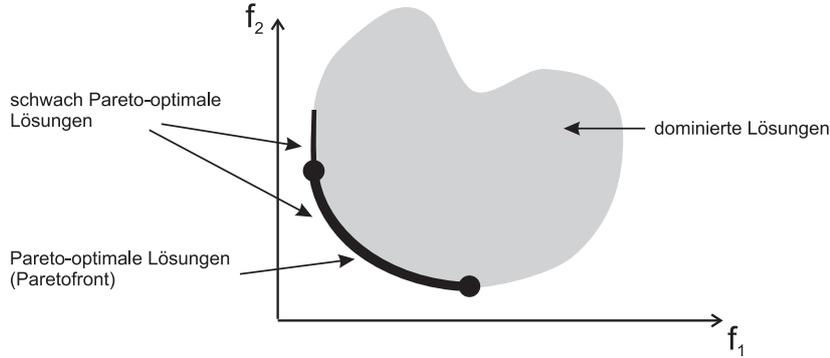


ABBILDUNG 1.1: Schematische Darstellung verschiedener Eigenschaften der Lösungsmenge eines bikriteriellen Problems. Insbesondere sind die dominierten Lösungen, die schwach Pareto-optimale Lösungen und die Paretofront abgebildet. Die schwach Pareto-optimale Lösungen zeichnen sich dadurch aus, dass es zwar bzgl.  $f_2$  bessere Lösungen gibt, nicht aber bzgl.  $f_1$ , siehe Def. 1.3.

#### Definition 1.4 (Lokale Pareto-Optimalität)

Sei  $U_\delta(\vec{x}) := \{\vec{y} \in \mathbb{R}^n \mid \|\vec{x} - \vec{y}\|_2 < \delta\}$  eine  $\delta$ -Umgebung von  $\vec{x} \in S \subseteq \mathbb{R}^n$ . Wir nennen  $\vec{x} \in S$  lokal Pareto-optimal (bzgl.  $f$ )  $:\Leftrightarrow \exists \delta > 0 : \vec{x} \in \mathcal{X}_f^*|_{U_\delta(\vec{x}) \cap S}$ .

Entsprechend unserer Notation schreiben wir:

$$\mathcal{X}^{*,loc} := \mathcal{X}_f^{*,loc} := \{\vec{x} \in S \mid \vec{x} \text{ lokal Pareto-optimal bzgl. } f\}.$$

#### Bemerkung 1.5 (Einkriterieller Fall)

Unter der Bedingung  $m = 1$  gilt:

- $\mathcal{X}_f^* = \{\text{globale Minima von } f\}$
- $\mathcal{X}_f^{*,loc} = \{\text{lokale Minima von } f\}$

#### Definition 1.6 (Mehrkriterielles Optimierungsproblem)

Gegeben sei  $f : S \rightarrow Z$ ,  $S \subseteq \mathbb{R}^n$ ,  $Z \subseteq \mathbb{R}^m$ . Das mehrkriterielle Optimierungsproblem (MOP) ist die Aufgabe, die Menge  $\mathcal{X}_f^*$  zu bestimmen.

Abschließend definieren wir nun noch die Konvexität eines mehrkriteriellen Optimierungsproblems und zeigen eine wichtige Eigenschaft konvexer MOPs in Verbindung mit Definition 1.4.

#### Definition 1.7 (Konvexität eines MOP)

Sei  $S \subseteq \mathbb{R}^n$  gegeben. Wir definieren folgende Zusammenhänge:

- Die Menge  $S$  heißt konvex, wenn für alle  $\vec{x}, \vec{y} \in S$  und alle  $\beta \in [0, 1]$  gilt  $\beta\vec{x} + (1 - \beta)\vec{y} \in S$ .
- Eine Funktion  $h : S \rightarrow \mathbb{R}$  heißt konvex, wenn  $S$  konvex ist und für alle  $\vec{x}, \vec{y} \in S$  und  $\beta \in [0, 1]$  gilt  $h(\beta\vec{x} + (1 - \beta)\vec{y}) \leq \beta h(\vec{x}) + (1 - \beta)h(\vec{y})$ .

- Ein mehrkriterielles Optimierungsproblem  $f$  wird als konvex bezeichnet, wenn die Komponentenfunktionen  $f_1, \dots, f_m$  konvex sind.

**Satz 1.8 (Globale Optimalität)**

Das MOP sei konvex. Jede lokal Pareto-optimale Lösung des MOP ist dann ebenfalls global Pareto-optimal:  $\mathcal{X}_f^{*,loc} = \mathcal{X}_f^*$ .

**Beweis:** Sei  $\bar{x}^* \in S$  lokal optimal. Es gibt also nach Definition 1.4 ein  $\delta > 0$ , so dass in  $U_\delta(\bar{x}^*) \cap S$  kein  $\bar{x}$  zu finden ist, welches  $\bar{x}^*$  echt dominiert. Wir formulieren weiterhin folgende Annahme:  $\bar{x}^*$  ist nicht global optimal, es gibt also ein  $\bar{x}' \in S$  mit  $\bar{x}' \prec \bar{x}^*$ .

Sei nun  $\bar{x}'' = \beta\bar{x}' + (1 - \beta)\bar{x}^*$  mit  $0 < \beta \leq 1$  derart, dass  $\bar{x}''$  in  $U_\delta(\bar{x}^*) \cap S$  liegt. Aufgrund der Konvexität aller Funktionen  $f_i$ ,  $1 \leq i \leq m$  gilt:

$$f_i(\bar{x}'') = f_i(\beta\bar{x}' + (1 - \beta)\bar{x}^*) \leq \beta f_i(\bar{x}') + (1 - \beta)f_i(\bar{x}^*)$$

Nach Konstruktion ist  $f_i(\bar{x}') \leq f_i(\bar{x}^*)$  für alle  $i$  und  $f_i(\bar{x}') < f_i(\bar{x}^*)$  für mindestens ein  $i$ , also:

$$f_i(\bar{x}'') \leq f_i(\bar{x}^*) \text{ für alle } i.$$

und

$$f_i(\bar{x}'') < f_i(\bar{x}^*) \text{ für mindestens ein } i, \text{ d.h. } \bar{x}'' \prec \bar{x}^*.$$

Da  $\bar{x}'' \in U_\delta(\bar{x}^*) \cap S$ , steht dies im Widerspruch zu der Voraussetzung, dass  $\bar{x}^*$  lokal Pareto-optimal ist. □

Nachdem nun mehrkriterielle Optimierungsprobleme formal definiert und ihre wichtigsten Eigenschaften beschrieben wurden, soll der zweite Teil dieses Kapitels kurz die Unterscheidung von analytischen und numerischen Methoden ansprechen. Auf einige wichtige Ergebnisse dieser Darstellung wird im Verlauf der Arbeit zurückgegriffen.

## 1.2 Prinzipien von Optimierungsmethoden

Bereits einleitend ist die Unterscheidung zwischen analytischen und numerischen Optimierungsmethoden als Möglichkeit der impliziten Klassifikation von Optimierungsproblemen angesprochen worden. Diese richtet sich grundsätzlich nach zwei Prinzipien der Ermittlung von optimalen Lösungen: Die analytischen Methoden versuchen in einem einzigen Schritt und durch die indirekte Ausnutzung spezieller Eigenschaften der Zielfunktionen mittels mathematischer Lösungsverfahren eine optimale Lösungen zu bestimmen. Hingegen nähern sich numerische Methoden schrittweise der Lösung an. Diese Verschiedenheit im Vorgehen begründet auch ihre alternativen Bezeichnungen als indirekte bzw. direkte Methoden.

### 1.2.1 Analytischer Ansatz

An dieser Stelle sollen kurz einige Aspekte indirekter Optimierungsmethoden diskutiert und zugleich zwei für diese Arbeit wichtige Eigenschaften von Pareto-optimalen Lösungen genauer betrachtet werden.

Die analytische Betrachtung von Optimierungsproblemen basiert immer auf der Untersuchung von Eigenschaften der Zielfunktion(en), um in einem einzigen Analyseschritt möglichst alle Optima festzustellen und zugleich deren lokale oder globale Optimalität nachzuweisen. Liegt im Falle der einkriteriellen Optimierung eine stetig differenzierbare Funktion  $f(\vec{x})$  einer oder mehrerer Veränderlicher vor, so können ein notwendiges und ein hinreichendes Kriterium – basierend auf partiellen Ableitungen  $\nabla f(\vec{x}) = \left( \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right) (\vec{x})$  der Funktion – für die Optimalität von Entscheidungsvektoren formuliert werden: Notwendig für (lokale) Optimalität an der Stelle  $\vec{x}$  ist, dass der Gradient  $\nabla f(\vec{x})$  verschwindet, also in jeder Komponente den Wert 0 annimmt. Eine hinreichende Bedingung zur Bestätigung der Optimalität eines Lösungskandidaten  $\vec{x}$  liefert z.B. die Definitheit<sup>7</sup> der Hesse-Matrix  $H(f(\vec{x})) = \left( \frac{\partial^2 f}{\partial x_i \partial x_j} \right)_{1 \leq i, j \leq n} (\vec{x})$ . Dies setzt jedoch voraus, dass die gegebene Funktion mindestens zweimal differenzierbar ist. Da diese Anforderung sehr stark und in der Realität oft nur eine Approximation des Gradienten verfügbar ist, kann zumindest auf die notwendige Bedingung zurückgegriffen werden. Die Kenntnis über die Konvexität eines Problems kann dann sogar dazu führen, dass wir eine hinreichende Aussage zur globalen Optimalität treffen können.

Die folgende notwendige Bedingung von John (1948) verallgemeinert das eben besprochene Verschwinden des Gradienten für ein Kriterium  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  und Randbedingungen  $g_1(\vec{x}), \dots, g_k(\vec{x})$  mit  $g_i : \mathbb{R}^n \rightarrow \mathbb{R}, g_i(\vec{x}) \leq 0$  für  $1 \leq i \leq k$ .

**Satz 1.9 (Fritz John's notwendige Bedingung)**

Seien  $f$  und  $g_1, \dots, g_k$  einmal stetig differenzierbare Funktionen und sei  $\vec{x}^*$  ein lokales Minimum von  $f$  unter den gegebenen Beschränkungen  $g_i(\vec{x}) \leq 0$ . Dann gibt es ein  $(\nu_0, \nu_1, \dots, \nu_k) \neq \vec{0}$  mit  $0 \leq \nu_i \in \mathbb{R}$ , so dass

- $\nu_0 \nabla f(\vec{x}^*) + \sum_{i=1}^k \nu_i \nabla g_i(\vec{x}^*) = 0$
- $\nu_i g_i(\vec{x}^*) = 0$  für alle  $1 \leq i \leq k$

Anschaulich bedeutet dies, dass unter der Annahme eines lokalen Optimums, zwei Fälle betrachtet werden können:

1.  $\nabla f(\vec{x}^*) = 0$  mit  $0 < \nu_0$  gilt und die Gewichte der Randbedingungen können als  $\nu_i = 0$  für  $1 \leq i \leq k$  gewählt werden.
2. Einige Randbedingungen sind mit  $g_i(\vec{x}^*) = 0, i \in 1, \dots, k$  aktiv, so dass einige  $0 < \nu_i$  und insbesondere  $\nu_0 = 0$  gewählt werden können. Die  $0 < \nu_i$

---

<sup>7</sup>Eine positiv definite Matrix bestätigt ein Minimum, negative Definitheit ein Maximum. Bei einer indefiniten Matrix liegt ein Sattelpunkt vor, während bei Semi-Definitheit keine Aussage getroffen werden kann.

können dann so gewählt werden, dass die gewichtete Summe der Gradienten der entsprechenden  $g_i(\vec{x}^*)$  verschwindet.

Ein Beweis dieses Satzes findet sich bei Simon (1976). Im Falle ohne Randbedingungen ergibt sich eben wieder die bereits oben diskutierte notwendige Bedingung. Fritz John's Satz lässt sich auch auf den mehrkriteriellen Fall übertragen und wie folgt formulieren.

**Satz 1.10 (Fritz John's notwendige Bedingung für Pareto-Optimalität)**

Seien  $f_1, \dots, f_m, g_1, \dots, g_k$  einfach stetig differenzierbar auf dem  $\mathbb{R}^n$  und zudem  $\vec{x}^*$  unter den Randbedingungen ein (lokales) Pareto-Optimum für  $f = (f_1, \dots, f_m)$ . Dann gibt es ein  $((\nu_1, \dots, \nu_m), (\omega_1, \dots, \omega_k)) \neq (\vec{0}, \vec{0})$  mit  $0 \leq \nu_i, \omega_i$ , so dass

- $\sum_{i=1}^m \nu_i \nabla f_i(\vec{x}^*) + \sum_{i=1}^k \omega_i \nabla g_i(\vec{x}^*) = 0$
- $\omega_j g_j(\vec{x}) = 0$  für  $1 \leq j \leq k$

Betrachten wir nur unbeschränkte Probleme, so entfallen die Randbedingungen und wir können feststellen, dass sich in einem (lokalen) Pareto-Optimum, bei geeigneter Gewichtung, alle Gradienten gegeneinander aufheben. Kuhn & Tucker (1951) haben gezeigt, dass diese Eigenschaft im Falle einer konvexen Problemstellung ebenfalls hinreichend für globale Pareto-Optimalität ist. Diese wichtige Eigenschaft können wir später für einen lokalen Suchmechanismus nutzen. Hier nutzen wir sie, um beispielhaft eine analytische Lösung für ein später häufig genutztes zweikriterielles Problem zu ermitteln.

**Beispiel 1.11 (Mehrkriterielle Kugelfunktion)**

Es sei das folgende mehrkriterielle Optimierungsproblem gegeben:

$$\mathcal{F}_{\odot}^m : \mathbb{R}^n \rightarrow \mathbb{R}^m \text{ mit } \vec{x} \in \mathbb{R}^n, m, n \in \mathbb{N}$$

$$\mathcal{F}_{\odot}^m(\vec{x}) = \begin{pmatrix} (\vec{x} - \vec{c}_1)^2 \\ \vdots \\ (\vec{x} - \vec{c}_m)^2 \end{pmatrix}, \vec{c}_i \in \mathbb{R}^n, \text{ konstant} \quad (1.2)$$

$$\vec{c}_i \neq \vec{c}_j \text{ für } i \neq j$$

Formuliert man eine einfache zweikriterielle Instanz des Problems mit  $\vec{c}_1 = (0, 0)^T$  und  $\vec{c}_2 = (2, 0)^T$ , so ergibt sich:

$$\mathcal{F}_{\odot}^2(\vec{x}) = \begin{pmatrix} x_1^2 + x_2^2 \\ (x_1 - 2)^2 + x_2^2 \end{pmatrix} \quad (1.3)$$

Für dieses einfache Problem soll die Paretomenge beispielhaft auf analytischem Wege bestimmt werden. Mit zwei Kriterien ergibt sich

$$\nu_1 \nabla f_1(\vec{x}) + \nu_2 \nabla f_2(\vec{x}) = 0$$

$$\Rightarrow -\nu \nabla f_1(\vec{x}) = \nabla f_2(\vec{x}), \text{ mit } \nu = \frac{\nu_1}{\nu_2} \text{ und o.E. } \nu_2 \neq 0$$

als Spezialfall des Satzes 1.10 von Fritz John und drückt aus, dass die Gradienten bezüglich beider Kriterien (anti-)parallel sind,<sup>8</sup> wenn ein Pareto-optimaler Punkt vorliegt. Es ist also die Menge von Lösungen für das folgende Gleichungssystem zu bestimmen, um direkt die Paretomenge  $\mathcal{X}^*$  zu erhalten:

$$-\nu \begin{pmatrix} 2x_1 \\ 2x_2 \end{pmatrix} = \begin{pmatrix} 2(x_1 - 2) \\ 2x_2 \end{pmatrix}$$

$$\Leftrightarrow -2x_1\nu = 2x_1 - 4 \tag{1.4}$$

$$-2x_2\nu = 2x_2 \tag{1.5}$$

Für (1.4) ergibt sich dann:

$$2x_1 + 2x_1\nu = 4 \Leftrightarrow x_1 = \frac{2}{1 + \nu}$$

und für (1.5):

$$2x_2 + 2x_2\nu = 0 \Leftrightarrow x_2 = 0$$

Für die Paretomenge ergibt sich hieraus für  $\nu \geq 0$ :

$$\{(x_1, 0) \mid x_1 \in ]0, 2]\} \subseteq \mathcal{X}^*.$$

Da  $x_1 = 0$  globales Minimum von  $f_1$  ist, ist auch  $(0, 0) \in \mathcal{X}^*$ . Die Paretofront ergibt sich wegen  $x_2 = 0$  als Funktion

$$f_2(f_1) = f_1 - 4\sqrt{f_1} + 4.$$

Abbildung 1.2 zeigt sowohl Paretomenge wie Paretofront des betrachteten Beispiels.

□

Die analytische Lösung von Optimierungsproblemen scheitert jedoch, wenn die Zielfunktionen nicht mehr stetig differenzierbar sind. Aber auch im Falle von Zielfunktionen höherer Ordnung kann eine schrittweise Annäherung an das Optimum effizienter sein, als eine Lösung mit analytischen Verfahren zu realisieren. Daher betrachtet der folgende Abschnitt nun die numerischen Methoden zur Bestimmung eines Optimums und leitet damit zu den in dieser Arbeit eingesetzten evolutionären Verfahren (siehe Kapitel 2) über.

### 1.2.2 Numerischer Ansatz

Die grundlegende Idee der numerischen Lösung von Optimierungsproblemen liegt in der schrittweisen Annäherung an die optimale Lösung. Dieses Vorgehen unterscheidet Schwefel (1995) in seiner umfassenden Darstellung wiederum grob in zwei

<sup>8</sup>Peschel (1980) bezeichnet dies in seinem Buch als Stationaritätsbedingung.

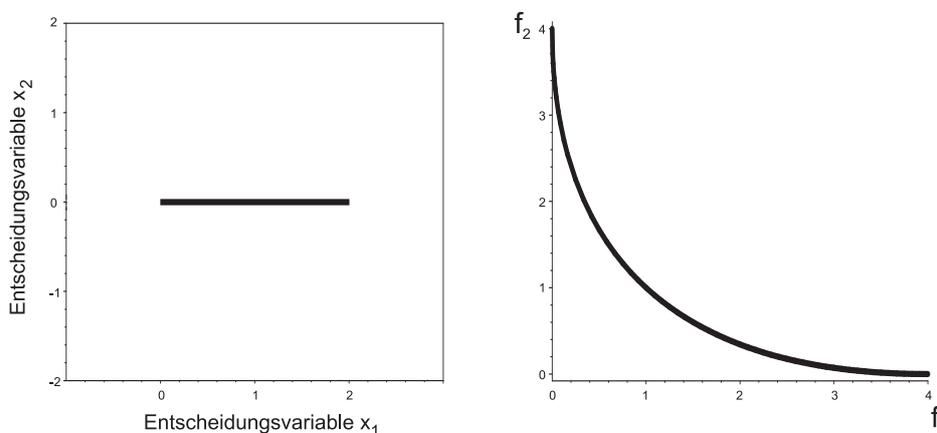


ABBILDUNG 1.2: Darstellung der analytisch bestimmten Paretomenge und Paretofront des Beispielproblems  $\mathcal{F}_\odot$ .

Klassen: Verfahren, die ein Modell funktionaler Abhängigkeiten zwischen Parametern und Systemzuständen zugrunde legen und jene, die ohne jegliche Modellannahme arbeiten. Als Vertreter der ersten Klasse beschreibt er verschiedene Interpolationsverfahren: Die Regula falsi-Methode beruht auf einem linearen Modell für die Zielfunktion, während das Newton-Raphson-Verfahren ein quadratisches Modell annimmt und das Lagrange-Verfahren  $p$  Stützstellen zur Interpolation eines Polynoms  $p$ -ten Grades nutzt. Alle Verfahren nutzen das unterliegende Modell, um von einem Ausgangspunkt der Suche aus einen weiteren Punkt näher am Optimum zu finden.

In ähnlicher Weise verfahren auch direkte Suchmethoden. Sie verzichten aber im Allgemeinen auf die Annahme eines zugrunde liegenden Modells. Vielmehr betrachten sie die direkte Nachbarschaft eines Punktes im Suchraum und versuchen dort heuristisch oder schematisch eine Verbesserung bezüglich der Zielfunktion zu erreichen. Auch für diese Klasse stellt Schwefel in seinem Buch eine Vielzahl von wichtigen Ansätzen vor<sup>9</sup>.

Für beide Verfahrensklassen, mit und ohne Modell, kann eine allgemeine Iterationsvorschrift angegeben werden, siehe Iterationsvorschrift (1.6). In dieser Darstellung ist  $\vec{x}_{k+1} \in \mathbb{R}^n$  ein neuer Punkt im Entscheidungsraum, der sich aus dem vorhergehenden Punkt  $\vec{x}_k \in \mathbb{R}^n$  und einem gerichteten Schritt  $\vec{s} \in \mathbb{R}^n$  ergibt.

$$\vec{x}_{k+1} = \vec{x}_k + \vec{s} \quad (1.6)$$

Der gerichtete Schritt lässt sich wiederum zerlegen in  $\vec{s} = \sigma_k \cdot \vec{e}_k$  mit  $\sigma_k \in \mathbb{R}$  als skalare Schrittweite und  $\vec{e}_k \in \mathbb{R}^n$  als Einheitsrichtungsvektor. Damit stellt sich ein numerisches Suchverfahren als die Aufgabe dar, für einen Ausgangspunkt eine

<sup>9</sup>Auf eine Vorstellung wird an dieser Stelle verzichtet, da keines der Verfahren in dieser Arbeit genutzt wird. Trotzdem ist ein Studium der deterministischen und teilweise sehr ausgefeilten Methoden sowohl als Inspiration wie auch als wichtige und oft vorzuziehende Alternative zu randomisierten Verfahren ratsam.

geeignete Suchrichtung und Schrittweite für einen gewinnbringenden – d.h sich dem Optimum annähernden – Schritt im Entscheidungsraum zu finden.

Die allgemeine Iterationsvorschrift gilt sowohl für Verfahren der einkriteriellen wie der mehrkriteriellen Optimierung, da sie sich ausschließlich auf den Entscheidungsraum bezieht. Jedoch kommen iterative Verfahren nicht ohne eine *Bewertung* im Zielfunktionsraum aus. Nach jedem Iterationsschritt muss der Gewinn, also die Annäherung an die Zielsetzung, überprüft werden. Hierbei muss dann natürlich wieder zwischen ein- und mehrkriteriellem Fall unterschieden werden. Ist ein Vergleich von Lösungsqualität im einkriteriellen Fall aufgrund der dort gegebenen Dominanzrelation problemlos möglich, stellt die Unvergleichbarkeit von Lösungen im Mehrkriteriellen direkt ein Problem für die Bewertung einer neuen Lösung dar. Eine einfache Übertragung einkriterieller Verfahren ist also nicht möglich.

Trotz der Bewertungsproblematik stellt das allgemeine Vorgehen der direkten Suche ein einfaches Modell für einen natürlichen Prozess dar, der Grundlage für vielfältige Verfahrensentwicklung und auch Gegenstand dieser Arbeit ist. Die sich wiederholende Folge von Lösungsvariation und Lösungsselektion kann unter dem Einfluss einer Zufallskomponente als sehr vereinfachte Repräsentation eines Evolutionsprozesses betrachtet werden, der von Darwin (1859) als „Überleben des besser Angepassten“ bezeichnet wird. Ende der 1960er wurde dieses Prinzip annähernd gleichzeitig von Holland (1975) und Fogel, Owens & Walsh (1966) in diskreten Entscheidungsräumen sowie von Rechenberg (1971) und Schwefel (1968) in reellwertigen Parameterräumen auf die Lösung von Optimierungsproblemen übertragen. Im Gegensatz zu damals etablierten Verfahren setzten die Autoren auf zufällige Einflüsse bei der Richtungswahl und eine Population von Lösungen. Eine grundlegende Betrachtung sowohl einkriterieller wie auch mehrkriterieller evolutionärer Verfahren erfolgt im nächsten Kapitel als Ausgangspunkt für die Einordnung des in dieser Arbeit untersuchten Verfahrens.



## Kapitel 2

# Evolutionäre Verfahren

Die biologische Evolution gilt seit Darwin (1859) als weithin anerkanntes und grundlegendes Erklärungsmodell für die Entwicklung und Weiterentwicklung von Lebewesen. Dabei werden unter gegebenen Umweltbedingungen nachweislich optimale Lösungen etwa für vielfältige Konstruktionsprobleme gefunden<sup>1</sup>, die sich nicht selten auch in der technischen Übertragung der Bionik als idealer Weg zur Problemlösung erweisen (Nachtigall, 2002).

Abhängig von der natürlichen Umgebung wird ein Entwicklungsdruck auf Spezies ausgeübt, der mit der Zeit zur Anpassung der Spezies als Ganzes und nur dadurch zur Sicherung ihres Fortbestandes führt. Diese Anpassung folgt einem Prinzip, das sich auf Populationen aus einzelnen sich zufällig verändernden Individuen gründet, deren genetische Anlagen sich durch Fortpflanzungserfolg aufgrund phänotypischer Überlegenheit in der gesamten Population verbreiten. Zufällige Mutationen und die Rekombination von Erbanlagen verschiedener Individuen spielen eine zentrale Rolle bei der Erzeugung von Innovation und Diversität im Phänotyp. Zugleich impliziert der Fortpflanzungserfolg eines Individuums einen Selektionsvorgang der über den Bestand und die Verbreitung der Neuerungen in der Population bestimmt.

Betrachtet man das Grundprinzip dieser Methodik bestehend aus sich wiederholenden Variationen (Mutation und Rekombination) und Selektionsvorgängen, so stellt man eine starke Ähnlichkeit zu der Iterationsvorschrift 1.6 numerischer Optimierungsmethoden in Kapitel 1 fest<sup>2</sup>. In dieser Weise auf ein Optimierungsproblem angewandt, kann im technischen Evolutionsprozess jedes Individuum vereinfacht als eine Lösung angesehen werden. Die Darstellung einer Lösung im Problemkontext wird als genetische Kodierung bezeichnet, während darauf durchgeführ-

---

<sup>1</sup>Nur einige Beispiele von durch Evolution entstandener optimaler Lösungen von Konstruktionsproblemen sind etwa die Querschnitte der Blutgefäße im menschlichen Körper, die Existenz hohler Knochenstrukturen bei Vögeln oder auch die selbstreinigende Oberflächenstruktur von Pflanzenblättern.

<sup>2</sup>Hier sei noch einmal explizit die einfache und ingenieurmäßige Übertragung des Evolutionsprinzips betont. Die Vergleichbarkeit mit den hochkomplexen natürlichen Anpassungsprozessen der biologischen Evolution soll hier nicht diskutiert werden.

te genetische Operationen dann Suchoperationen darstellen und einen Innovations-schritt herbeiführen. Abweichend von der deterministischen Vorgehensweise in Vorschrift 1.6 wird im Evolutionsprozess nicht generell von einem Fortbestand der neu generierten Lösung ausgegangen. Erst ein Selektionsmechanismus bestimmt, ob der Iterationsschritt durchgeführt wird. Analog zur biologischen Nomenklatur wird für das zeitliche Voranschreiten der Evolution auf den Generationenbegriff zurückgegriffen. Im Folgenden wird zuerst die allgemeine Methodik definiert und anschließend auf grundlegende Eigenschaften evolutionärer Algorithmen eingegangen.

## 2.1 Grundlagen und Prinzipien

Ende der Sechziger Jahre des vorherigen Jahrhunderts entwickelten sich unabhängig voneinander drei Arten evolutionärer Algorithmen, die zugleich auch unterschiedliche Aspekte der Optimierung betrachteten. Während Fogel et al. (1966) die evolutionären Prinzipien für die automatische Entwicklung endlicher Automaten<sup>3</sup> einsetzten und ihr Vorgehen evolutionäre Programmierung (EP) nannten, nutzte Holland (1975) sowie dessen Schüler De Jong (1975) und Goldberg (1983) ebenfalls die Naturanalogie, um kombinatorische Optimierungsprobleme zuerst auf der Basis binärer Kodierung zu lösen. Dieser Ansatz wurde unter dem Namen der genetischen Algorithmen (GA) bekannt und später auch auf ganzzahlige Repräsentationen übertragen (Michalewicz, 1999). Auch Schwefel (Schwefel, 1968, 1995) und Rechenberg (1971) wandten die aus der Biologie entlehnten Methoden an, spezialisierten sie jedoch zuerst auf die ganzzahlige Parameteroptimierung im ingenieurtechnischen Umfeld und verwendeten sie später in allgemeinerer Form in der reellwertigen Optimierung als sogenannte Evolutionsstrategien (ES). Inzwischen ist eine Unterscheidung der Methoden nicht mehr uneingeschränkt aufrecht zu erhalten. Frühere Abgrenzungen lassen sich nur noch zwischen EP und ES/GA finden, da die EP nach wie vor ausschließlich auf die Mutation als Variationsoperator setzt. Unterscheidungen zwischen ES und GA bezüglich der Notation, der Kodierung von Individuen und der Rangfolge der Variationsoperatoren haben an Bedeutung verloren<sup>4</sup>. Aus diesem Grund werden im Folgenden Evolutionsstrategien und genetische Algorithmen als verbreitete Repräsentanten der evolutionären Algorithmen zur ein- aber auch mehrkriteriellen Optimierung behandelt. Spezielle Aspekte werden an entsprechender Stelle kurz hervorgehoben.

---

<sup>3</sup>Ein endlicher Automat ist die Möglichkeit ein Verhaltensmodell, etwa den Ablauf von Programmen oder Berechnungsvorschriften abstrakt durch Zustände, Übergänge und Aktionen zu beschreiben, (engl. finite state automaton).

<sup>4</sup>Die ursprüngliche Unterscheidung besagt, dass genetische Algorithmen Individuen mit binärer oder ganzzahliger Kodierung verwenden und die Mutation eine der Rekombination (Cross-Over im Sprachgebrauch der GAs) nachgeordnete Rolle spielt. Die ES hingegen behandelt ursprünglich beide Operatoren gleich gewichtet und stellt damit einen Mittelweg zwischen EP und GA dar.

## 2.2 Notation und Evolutionsschleife

Der übliche Ausgangspunkt für die Beschreibung von evolutionären Ansätzen ist die Darstellung der sogenannten Evolutionsschleife und der Notation zur Unterscheidung verschiedener Verfahrensausprägungen. Da die evolutionären Algorithmen zu den numerischen Methoden gehören, lässt sich die Evolutionsschleife, also der sich wiederholende Ablauf der Weiterentwicklung von Individuen, direkt über Iterationsvorschrift 1.6, Kapitel 1 darstellen. Ein neues Individuum entsteht, indem ein sogenanntes Elternindividuum repliziert und leicht verändert wird. Diese Veränderung geschieht durch einen Variationsoperator, der im einfachsten Fall eine zufällige Veränderung der genetischen Information des Elternindividuums herbeiführt. Ein Selektionsmechanismus überprüft anschließend, ob die Veränderung ein Nachkommenindividuum erzeugt hat, das im Rahmen der Bewertungsfunktion (Fitnessfunktion) seinem Elternteil in der Fitness gleicht oder ihn übertrifft. Ist dies der Fall, so bildet das Nachkommenindividuum den Ausgangspunkt für den nächsten Durchlauf der Evolutionsschleife. Andernfalls wird das Elternindividuum als momentan beste Lösung beibehalten.

Die dargestellte Verfahrensweise betrachtet den einfachsten Fall einer Eltern-Nachkommen-Beziehung. Allgemein treten im Evolutionsprozess der ES  $\mu$  Elternindividuen und  $\lambda$  Nachkommen dieser Eltern auf, aus denen durch Selektion eine Population von Lösungen für die nächste Generation gebildet wird. Die Menge der  $\lambda$  Nachkommenindividuen entsteht dabei immer durch die Anwendung von Variationsoperatoren auf die Elternindividuen. Die fundamentale Unterscheidung von Evolutionsstrategien wird jedoch über den Selektionsmechanismus getroffen. So entstehen die  $\mu$  Individuen der nächsten Generation in der Bandbreite einer elitären  $(\mu + \lambda)$ -Selektion und einer exklusiven  $(\mu, \lambda)$ -Selektion. Während bei der elitären Selektion die neuen Elternindividuen aus der Vereinigung der Eltern- und Nachkommenindividuen im Selektionsprozess entstehen, wird die exklusive Selektion ausschließlich auf der Menge der  $\lambda$  Nachkommenindividuen durchgeführt. Zwischenstufen der Selektionsmechanismen werden durch die Einführung einer maximalen Lebenszeit  $\kappa$  (in aufeinander folgenden Generationen) für Individuen erreicht und mit  $(\mu, \kappa, \lambda)$  notiert (Schwefel, 1995).

Der Unterschied beider Extremfälle liegt in der Überlebensmöglichkeit für Individuen und dem damit verbundenen Systemverhalten. Während bei der  $(\mu + \lambda)$ -Selektion Individuen theoretisch unbegrenzt überleben können, überlebt ein Individuum bei Anwendung der  $(\mu, \lambda)$ -Selektion maximal einen Selektionsprozess. Im ersten Fall sichert die elitäre Methode das Überleben der bisher besten Lösung, kann aber zu einer Stagnation der Annäherung an das globale Optimum führen. Diesem Effekt wird durch das implizite „Vergessen“ von Lösungen in der exklusiven Selektion entgegen gewirkt, was aber zur Divergenz des Lösungsverfahrens führen kann (Beyer & Schwefel, 2002).

Die einfachste Variante einer Evolutionsstrategie ist die zur Einführung genutzte  $(1+1)$ -ES. Im Laufe dieser Arbeit wird neben jener einfachen Variante die in der Literatur verbreitet als Steady-State-ES bezeichnete  $(\mu + 1)$ -ES verwendet (Beyer

& Schwefel, 2002). Bei diesem Ansatz wird ausgehend von  $\mu$  Elternindividuen nur ein Nachkomme gebildet, der anschließend ein Individuum aus der Elternpopulation ersetzen kann. Verbreitet ist dieser Ansatz insbesondere für die Umsetzung sogenannter zellulärer evolutionärer Algorithmen (engl.: cellular evolutionary algorithms), in deren Umsetzung die Population eine räumliche Struktur aufweist und nur eine lokale Nachbarschaft für die Durchführung des Evolutionsschrittes betrachtet wird, siehe auch Abschnitt 2.2.4.

Ausgehend von den grundlegenden Eigenschaften und der vorgestellten Notation der ES werden in den folgenden Abschnitten einige spezielle Aspekte betrachtet. Ein besonderer Fokus liegt dabei auf der Populationsstruktur, der Repräsentation von Individuen sowie den Variationsoperatoren, den Elementen also, die im Laufe der Arbeit verstärkt Beachtung finden.

### 2.2.1 Repräsentationen von Individuen

Individuen stellen in einem evolutionären Algorithmus Lösungskandidaten für das betrachtete Optimierungsproblem dar und enthalten jeweils individuelle und vollständige Informationen über die Lösung in Form von Informationsbausteinen. Die analogen biologischen Grundlagen hat Mendel (1865) bereits früh mit Pflanzenzuchtversuchen gelegt, die aufzeigten, dass jedes Individuum genetische Informationen enthält, welche seine Form und Fähigkeiten entscheidend bestimmen. Dabei wird implizit eine Unterscheidung von genetischer Repräsentation (Genotyp) und äußerer Erscheinung (Phänotyp) vorgenommen. Ein Genotyp stellt also die interne Repräsentation aller das Individuum betreffenden Informationen dar, während der Phänotyp letztlich die Umsetzung der Informationen – gleichsam wie durch eine Transformationsfunktion – in ein Erscheinungsbild des Individuums beschreibt. Für das Überleben eines Individuums in einer Umgebung und damit für den Vergleich verschiedener Individuen ist somit der Phänotyp entscheidend, der von der Art der Expression des Genotyps abhängt. Für die Reproduktion und Weitergabe von Eigenschaften eines Individuums muss hingegen der Genotyp betrachtet werden (Rothlauf, 2006).

Formal kann daher bei der Repräsentation von Individuen für einen evolutionären Algorithmus zwischen einem genotypischen Suchraum  $\Phi_g$  und einem phänotypischen Suchraum  $\Phi_p$  unterschieden werden und gleichzeitig die Zielfunktion in zwei Abbildungen zerlegt werden, siehe Gleichungen 2.1 und 2.2.

$$f_g : \Phi_g \rightarrow \Phi_p \tag{2.1}$$

$$f : \Phi_p \rightarrow \mathbb{R} \tag{2.2}$$

Dabei stellt die erste Abbildung den Zusammenhang zwischen den genotypischen Informationen und der phänotypischen Repräsentation eines Individuums dar. Die zweite Abbildung drückt die Qualität des Phänotypen aus und ist Grundlage für die Selektion.

Für ein betrachtetes Optimierungsproblem kann die Abbildung der genotypischen auf die phänotypische Repräsentation ganz unterschiedlich gewählt werden.

Ronald (1997) weist darauf hin, dass die Wahl einer Repräsentation oftmals entscheidend darüber bestimmt, wie leistungsfähig ein evolutionärer Algorithmus auf der Population arbeitet. Praktisch relevante Aspekte sind dabei die Repräsentation von Teilen des Gesamtproblems in der Kodierung, damit zusammenhängend die Anwendbarkeit von Variationsoperatoren für die Weitergabe von Teillösungen und eine effiziente Übertragbarkeit des Genotyps in seine phänotypische Ausprägung. In diesem Kontext sind die Wahl einer geeigneten Abstraktionsebene für die Repräsentation des Problems, die Unterdrückung isomorpher Ausprägungen<sup>5</sup> und die Verwendung eines möglichst kleinen Alphabets für die Belegung eines Allels<sup>6</sup> ebenfalls hervorzuheben. Im *Handbook of Evolutionary Computation* (Bäck, Fogel & Michalewicz, 1997) werden zahlreiche Repräsentationsvarianten vorgestellt. An dieser Stelle seien jedoch nur die drei Ausprägungen erwähnt, die im Verlaufe der Arbeit genutzt werden oder auf die an einigen Stellen verstärkt verwiesen wird.

**Binäre und ganzzahlige Repräsentationen** sind insbesondere bei ganzzahligen Problemstellungen und damit in der Domäne der genetischen Algorithmen die häufigste Methode, Individuen auf genotypischer Ebene zu repräsentieren. Im binären Fall wird das Individuum durch einen String  $\{0, 1\}^l$  der Länge  $l$  repräsentiert. Bei ganzzahligen Repräsentationen besteht das Alphabet aus mehr als zwei Zeichen. Gewöhnlich wird die binäre oder ganzzahlige Repräsentation im Kontext von genetischen Algorithmen benutzt. So werden auch reellwertige Probleme in eine binäre Kodierung übertragen und mit den Mitteln genetischer Algorithmen gelöst. Als weiteres Beispiel für die Verwendung einer ganzzahligen Repräsentation im Sinne der Abbildungsvorschrift 2.1 sei auf die Arbeit von Grimme, Lepping & Papaspyrou (2008a) verwiesen. Die Autoren kodieren die Lösungskandidaten als Zeichenkette über dem Alphabet  $\{0, 1\}$ , um damit eine Partition von Rechenaufgaben (Jobs) auf zwei Rechensysteme (Sites) darzustellen. Ausgehend von einer gegebenen Sequenz von Aufgaben soll entschieden werden, welche Aufgabe auf Site  $S_1$  und welche Aufgabe auf Site  $S_2$  ausgeführt werden soll. Dabei kodiert ein String mit der Eingabelänge  $l$  der Jobsequenz die Zuweisung auf die Sites. Eine 0 an Position  $1 \leq i \leq l$  bedeutet die Zuweisung des Jobs  $i$  auf  $S_1$ . Entsprechend bedeutet eine 1 die Zuweisung des Jobs auf  $S_2$ . Durch die Aufnahme weiterer Symbole in das Alphabet lassen sich leicht umfangreichere Szenarien mit weiteren Sites darstellen.

**Permutationsrepräsentationen** folgen in der Regel aus ordnungsbasierten Problemstellungen wie dem Traveling Salesperson Problem<sup>7</sup> (TSP). Dabei muss

<sup>5</sup>Es soll verhindert werden, dass derselbe Phänotyp durch mehr als einen Genotyp ausgedrückt werden kann.

<sup>6</sup>Ein Allel wird im Kontext der evolutionären Algorithmen als die kleinste Informationseinheit innerhalb des Genoms gesehen. In der Regel handelt es sich dabei um eine Stelle im Genom, die mit einem Zeichen des Alphabets belegt werden kann.

<sup>7</sup>Das Problem beschreibt die Herausforderung für einen Handelsreisenden, eine Reihe von Städten in einer kosten-optimalen Rundtour so zu besuchen, dass jede Stadt genau einmal aufgesucht wird.

eine Menge von Symbolen (im Falle von TSP die Repräsentationen der zu besuchenden Städte) in eine optimale Ordnung gebracht werden. Bei dieser Repräsentation entspricht die Anzahl der verwendeten Allele genau der Anzahl der zu ordnenden Elemente. Kein Allel tritt in der Repräsentationszeichenkette doppelt auf. Mit dieser Besonderheit der Kodierung müssen auch die verwendeten Variationsoperatoren umgehen können, ggf. also Methoden zur Reparatur von doppelten Einträgen umfassen. Prominente Beispiele für diskrete Optimierungsprobleme die oftmals mittels einer Permutationskodierung dargestellt werden, sind Schedulingprobleme<sup>8</sup>, siehe Teil IV oder in der Literatur bei Pinedo (2008). Im letzten Teil der Arbeit wird auf diese Problemstellungen und ihre Kodierung detailliert eingegangen.

**Reellwertige Repräsentationen** werden vorwiegend verwendet, wenn der Suchraum reellwertig, also im  $\mathbb{R}^n$  definiert ist. Dabei stellt  $n$  die Länge des reellwertigen Vektors (der manchmal auch als Chromosom bezeichnet wird) und die Dimension des Suchraumes dar (Rothlauf, 2006). Die Nutzung reellwertiger Vektoren als Lösungskandidaten ist in der Entstehung der Evolutionsstrategien begründet. Wurden in ersten Experimenten noch diskrete Kontrollparameter verwendet, so überführte man diese für die Nutzung kontinuierlicher Zufallsverteilungen und für ein besseres und flexibleres Design von Variationsoperatoren in eine kontinuierliche Darstellung (Beyer & Schwefel, 2002). Allgemein stellt jede Komponente eines Lösungsvektors  $\vec{x} \in \mathbb{R}^n$  einen einstellbaren Parameter eines Systems oder einer Funktion dar. Zugleich bedeutet dies aber auch, dass es im Sinne von Abbildungsvorschrift 2.1 keine Unterscheidung zwischen genotypischer Darstellung und phänotypischer Ausprägung gibt.

### 2.2.2 Variationsoperatoren

Für die Exploration des Suchraumes und Generierung neuer Lösungskandidaten sind Variationsoperatoren sicherlich die wichtigsten Bestandteile eines evolutionären Algorithmus. Im Sinne von Iterationsvorschrift 1.6 steuert die Variation im Falle der Mutation, ausgehend von einer vorhandenen Lösung  $\vec{x}_k$ , den explorativen Suchschritt  $\vec{s}$  bei. Die Rekombination als weitere Klasse von Variationsoperatoren repräsentiert in Analogie zur biologischen Reproduktion die (sexuelle) Vermischung genetischer Informationen. Im Folgenden werden die wichtigsten Eigenschaften der Mutation und Rekombination dargestellt. Anschließend wird auf einige Ausprägungen der Operatoren eingegangen.

---

<sup>8</sup>Das Scheduling beschreibt dabei den Vorgang der Zuweisung von Aufgaben (Jobs) zu nur spärlich vorhandenen Ressourcen (Maschinen) über die Zeit. Dabei kann eine Lösung als Ordnung der zuzuweisenden Jobs und somit als Permutation der Jobnummern dargestellt werden.

### Mutation

Die Mutation wird biologisch als zufällige Veränderung eines Gens bezeichnet (Hennig, 1995) und bringt auch im evolutionären Algorithmus durch zufällige Veränderung von Lösungskandidaten Innovation in den Suchvorgang ein. Insgesamt ist die algorithmische Umsetzung dieses Explorationsmechanismus sehr abhängig von dem betrachteten Problem und der zugrunde liegenden Repräsentation eines Individuums, so dass nur wenige allgemeine Regeln für die Konstruktion festgelegt werden können. Beyer (2001) stellt im Rahmen der theoretischen Betrachtung von Evolutionsstrategien vier weitgehend allgemeine Aspekte heraus:

1. *Erreichbarkeit*: Ein Mutationsoperator muss die zufällige Variation derart einbringen, dass ein beliebiger Punkt im Suchraum von jedem anderen Punkt im Suchraum in endlicher Zeit erreicht wird. Diese Bedingung ist sozusagen die grundlegende Eigenschaft jeder Mutation und stellt eine prinzipielle Erkundung des gesamten Suchraumes sicher.
2. *Skalierbarkeit*: Der Einfluss der Mutation sollte veränderbar sein, um eine Anpassung an lokale Eigenschaften des Suchraums zu gewährleisten. Dies geschieht normalerweise über die Anpassung der sogenannten Mutations-schrittweite.
3. *Verzerrungsfreiheit*: Die zufällige Ermittlung der Suchrichtung soll keine Tendenz (engl. bias) in bestimmte Richtungen aufweisen, sondern jede Suchrichtung gleich wahrscheinlich erzeugen können.
4. *Symmetrie*: In Anlehnung an die vorherige Eigenschaft beschreibt Symmetrie die Umkehrbarkeit eines Mutationsübergangs mit gleicher Wahrscheinlichkeit. Die Transformation eines Elternindividuum in ein Nachkommenindividuum soll so wahrscheinlich wie der umgekehrte Prozess sein.

Nach Beyer stellen die vier Aspekte jedoch nur eine allgemeine Leitlinie für den Entwurf von Mutationen dar. Einzig die Erreichbarkeit jedes Punktes sollte sichergestellt sein, um in einer unbekanntem Fitnesslandschaft keine Lösung prinzipiell auszuschließen. Stehen jedoch Informationen über den Suchraum oder Wissen über die Ausprägung einer Lösung a priori zur Verfügung, so müssen sicherlich einige der zuvor genannten Aspekte relativiert werden. In diesem Falle kann die Vorgabe einer Suchrichtung durchaus zur Beschleunigung der Lösungsfindung beitragen. Bei reellwertiger Repräsentation der Individuen und einer unbekanntem Fitnesslandschaft wird jedoch allgemein auf eine normal verteilte Mutation  $\vec{z}$  wie in Gleichung 2.3 dargestellt zurückgegriffen.

$$\begin{aligned} \vec{x}_{k+1} &= \vec{x}_k + \vec{z}, & (2.3) \\ \text{mit } \mathbb{R} \ni \vec{x}_{k+1}, \vec{x}_k, \vec{z} &= (z_1, \dots, z_n)^T \\ \text{und } z_i &\sim \mathcal{N}(0, \sigma_i^2), i \in \{1, \dots, n\} \end{aligned}$$

Wird ein Individuum binär kodiert, so wird die Mutation anders definiert (Michalewicz, 1999): im einfachsten Fall hat jede Position im Genom die Chance, mutiert zu werden. Die Größe dieser Chance hängt direkt mit der Mutationsstärke  $\sigma_m$  zusammen. So ergibt sich die formale Mutationsvorschrift in Gleichung 2.4 für einen Binärstring der Länge  $l$ .

$$\begin{aligned} \vec{x}_{k+1} &= \text{mut}(\vec{x}_k, \sigma_m), \\ &\text{mit } \vec{x}_{k+1}, \vec{x}_k \in \{0, 1\}^l \text{ und } \sigma_m \in [0, 1] \end{aligned} \quad (2.4)$$

Dabei beschreibt die Funktion *mut* die Betrachtung jeder Position  $i \in \{1, \dots, l\}$  im Genom: jedes Mal wird eine Zufallszahl  $p_i = \mathcal{U}(0, 1)$  gleichverteilt gezogen und mit der Mutationsstärke verglichen. Trifft die Bedingung  $p_i < \sigma_m$  zu, so wird das Allel an der betreffenden Position durch das komplementäre Element ersetzt. Im Falle einer Permutationskodierung verändert sich neben der Darstellung des Genoms auch die Mutationsfunktion. Da jedes Allel nur einmal auftreten darf, beschränkt sich die Mutation auf die zufällige Veränderung der Ordnung. Den einfachsten Fall eines Mutationsschrittes stellt die sogenannte *Swap Mutation* dar. Hier werden zwei Positionen zufällig gewählt und die enthaltenen Allele vertauscht.

Die vorgestellten grundlegenden Mutationen erfüllen alle die oben angegebenen Voraussetzungen. Doch bereits die Betrachtung unterschiedlicher Kodierungen von Lösungskandidaten zeigt auf, dass Mutationen ganz unterschiedliche und oft auch problemspezifische Ausprägungen haben können.

An dieser Stelle soll nur der Vollständigkeit halber auf Verfeinerungen der Mutation hingewiesen werden, die Anforderung 2 (Skalierbarkeit) bedient und eine Anpassung der Mutationsstärke erlauben, um die Konvergenz eines Verfahrens zu beschleunigen. Einen ersten empirischen Ansatz zur Anpassung der Mutationsschrittweite schlug Rechenberg (1994) mit der 1/5-Regel vor. Dabei wird hin und wieder die Erfolgswahrscheinlichkeit der Mutation über einige Generationen ermittelt und die Schrittweite entsprechend des Richtwertes 1/5 angepasst: liegt die Erfolgswahrscheinlichkeit der Mutation über 1/5, so wird die Schrittweite vergrößert; ist die Erfolgswahrscheinlichkeit geringer als 1/5, so wird die Schrittweite verringert. Der Wert von etwa 1/5 ergibt sich aus der kombinierten Untersuchung der maximalen Konvergenzgeschwindigkeit am Kugel- und Korridormodell<sup>9</sup>. Einen weiterentwickelten und selbstadaptiven Ansatz schlug Schwefel (1975) vor. Diese Methode nutzt den Evolutionsprozess selbst, um Strategieparameter für die Mutationsschrittweite für jedes Individuum bezüglich seiner Umgebung im Suchraum anzupassen. Damit ist das Überleben und Erlernen guter Strategieparameter implizit an die Entwicklung jedes Individuums gebunden und erheblich flexibler als

---

<sup>9</sup>Diese beiden einfachen Zielfunktionen wurden für die Ermittlung der „optimalen“ Erfolgsquote von 1/5 für die Mutationsschrittweite genutzt. Das Kugelmodell besitzt dabei konzentrische Sphären als Konturen, das Korridormodell parallele Konturen, die in  $n - 1$  Dimensionen beschränkt sind und so einen Korridor bilden, siehe Schwefel (1995).

die von Rechenberg vorgeschlagene Methode<sup>10</sup>. Dies führt dazu, dass sie heute einen Quasistandard im Bereich der einkriteriellen Evolutionsstrategien für die Anpassung der Mutationsschrittweite darstellt.

### Rekombination

Im Gegensatz zur Mutation bringt die Rekombination keine neuen genetischen Informationen in die Population ein, sondern kombiniert Teilaspekte vorhandener Lösungen neu. Im biologischen Sinne stellt die Rekombination eine Vermischung des Erbguts zweier oder mehrerer Elternindividuen dar, beschreibt also im mathematischen Sinne eine konvexe Operation. Auch die Mechanismen der Rekombination sind stark von der Repräsentation der Individuen abhängig. Die Unterscheidung zwischen genetischen Algorithmen und Evolutionsstrategien ist daher auch historisch besonders an der Ausprägung der Rekombination und ihrer Bedeutung innerhalb der verschiedenen Methoden festzumachen. Während genetische Algorithmen die Rekombination (dort oft Crossover genannt) in Anlehnung an die biologische Analogie als zentralen Operator betrachten, ist sie für die Evolutionsstrategien in ihrer Bedeutung gleichwertig zum explorativen Charakter der Mutation. Hier spielt die Rekombination sowohl als ein die Varianz kontrollierender Operator, wie auch als wichtiger Einfluss für die Entwicklung der Strategieparameter der Mutationsschrittweite eine zentrale Rolle. Im Folgenden soll zuerst auf die Rekombinationsarten in reellwertigen Suchräumen eingegangen werden, da diese eine zentrale Rolle in der Arbeit spielen. Die anschließende Darstellung einiger Crossover-Operatoren für genetische Algorithmen und damit überwiegend ganzzahlige Kodierungen dient der Motivation späterer spezieller Ansätze.

Im reellwertigen Suchraum ergeben sich die Komponenten  $c(i)$  eines Nachkommens  $c$  durch Rekombination aus  $\rho$  Elternindividuen  $P = \{p_1, \dots, p_\rho\}$  mittels der Vorschriften (2.5) nach Schwefel & Rudolph (1995).

$$c(i) = \begin{cases} p_k(i), & \text{Diskrete Rekombination: } p_k \text{ zu-} \\ & \text{fällig gewählt aus } P. \\ \frac{1}{\rho} \sum_{k=1}^{\rho} p_k(i), & \text{Global intermediäre Rekombina-} \\ & \text{tion, alle Eltern tragen gleich bei.} \\ \alpha p_k(i) + (1 - \alpha) p_l(i) & \text{Lokal intermediäre Rekombina-} \\ & \text{tion: } p_k, p_l \in P \text{ für jeden Nach-} \\ & \text{kommen, wobei } \alpha \sim \mathcal{U}(0, 1) \\ & \text{gleichverteilt zwischen 0 und 1} \\ & \text{für jede Komponente gewählt} \\ & \text{wird.} \end{cases} \quad (2.5)$$

<sup>10</sup>Eine ausführliche Darstellung der Methode findet sich in Schwefels Buch „Evolution and Optimum Seeking“ (Schwefel, 1995). Zudem wird sie theoretisch ausführlich bei Beyer (2001), dort insbesondere in Kapitel 7, diskutiert.

Die diskrete Rekombination erstellt ein Nachkommenindividuum aus zufällig ausgewählten Komponenten der Elternindividuen. Dabei können ausschließlich Individuen an den „Ecken“ des aufgespannten Raums entlang der Koordinatenachsen des Suchraums entstehen, siehe  $c_1$  und  $c_2$  sowie  $p_1$  und  $p_2$  in Abbildung 2.1(A). Bei der intermediären Rekombination kann ausschließlich ein mittleres Individuum entstehen, dass in allen Komponenten den Mittelwert aller beteiligter Elternindividuen darstellt, siehe  $c_3$  in Abbildung 2.1(A).

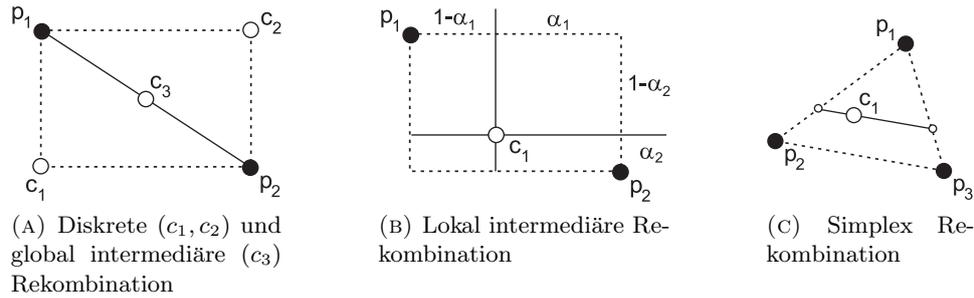


ABBILDUNG 2.1: Schematische Darstellung der diskreten, intermediären, lokal intermediären und Simplex-Rekombinationen.

Die lokal intermediäre Rekombination löst diese Beschränkung auf und erlaubt in jeder Komponente des Nachkommens eine Gewichtung der Elternkomponenten, begrenzt aber zugleich die Anzahl der Eltern auf zwei, siehe Abbildung 2.1(B). Diese Rekombinationsart ist, wie die vorherigen Varianten, ebenfalls entlang der Koordinatenachsen des Suchraumes definiert. Um diese Beschränkung zu überwinden, hat Grimme (2005) die sogenannte Simplex-Rekombination eingeführt, die analog zur lokal intermediären Rekombination gleichverteilt aus einem Simplex (im  $\mathbb{R}^2$  ein Dreieck) wählt. Dazu werden allgemein  $n + 1$  Eltern benötigt, mit  $n$  als Dimension des Suchraums. Die Methode beruht auf baryzentrischen Koordinaten und bestimmt einen Nachkommen über die in Gleichung 2.6 dargestellte Vorschrift. Eine detaillierte Herleitung findet sich in der erwähnten Arbeit von Grimme (2005).

$$c = \sum_{i=1}^{n+1} p_i \left( (1 - \lambda_i) \prod_{j=0}^{i-1} \lambda_j \right) \quad (2.6)$$

mit  $\lambda_0 := 1$  und  $\lambda_{n+1} := 0$

wobei für  $\lambda_i \in \{\lambda_1, \dots, \lambda_n\}$  gilt  $\lambda_i = (z_i)^{\frac{1}{(n+1)-i}}$

und  $z_i \sim \mathcal{U}(0, 1), i \in \{1, \dots, n\}$  gleichverteilt

Die spezielle Form zur Berechnung eines gleichverteilt zufälligen Punktes in einem Dreieck (siehe Abbildung 2.1(C)) lautet dann

$$c_1 = p_1(1 - \sqrt{z_1}) + p_2(1 - z_2)\sqrt{z_1} + p_3 z_2 \sqrt{z_1}$$

wobei  $z_1$  und  $z_2$  zufällig gleichverteilt zwischen 0 und 1 gewählt werden. Das Ziehen der Quadratwurzel (im allgemeinen der ersten bis  $n$ -ten Wurzel) geschieht hier, um die Proportionen des Simplex zu berücksichtigen und eine gleichverteilte Wahl des Nachkommenindividuum zu garantieren.

Die Vermischung von Informationen ganzzahlig repräsentierter Individuen wird im Bereich der genetischen Algorithmen als Crossover bezeichnet. Abbildung 2.2 zeigt das allgemeine Schema einer oftmals genutzten Methode für Binär- und Permutationskodierungen: für zwei Elternindividuen werden  $k$  sogenannte *cut points* festgelegt, an denen Informationen kreuzweise getauscht werden. Daraus entstehen zwei neue Nachkommenindividuen. Beispielhaft soll hier auf jeweils eine Aus-

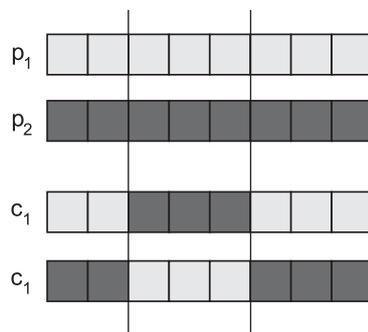


ABBILDUNG 2.2: Schematische Darstellung des Crossovers bei ganzzahliger Repräsentation mit zwei Cutpoints, an denen sich die Rekombination der genetischen Informationen ausrichtet.

prägung der Methode für binäre und permutationsbasierte Repräsentationen eingegangen werden, den  $k$ -Point Crossover (k-PX) und den von Davis (1985) vorgeschlagenen Order Crossover (OX). Der k-PX wählt  $k$  zufällige cut points für zwei binär kodierte Elternindividuen und führt dort jeweils zur Erstellung der Nachkommen eine Kreuzung der Informationen durch, wie in Abbildung 2.2 für  $k = 2$  dargestellt. In analoger Weise kann der OX ebenfalls arbeiten, muss dabei aber die Permutationskodierung beachten und sicherstellen, dass keine Dopplung der Allele geschieht. Dies wird so gelöst, dass zuerst die Teile aus  $p_1$  übernommen werden und anschließend die freien Stellen entsprechend der Ordnung der verbleibenden Allele in  $p_2$  eingefügt werden.

Insgesamt gibt es eine fast unüberschaubare Vielzahl von Crossover-Varianten. Etliche davon sind für die praktische Anwendung bei Michalewicz (1999) zu finden. Eine ausschließliche Aufzählung von Operatoren mit algorithmischer Beschreibung der Operatoren findet sich bei Gwiazda (2006).

### 2.2.3 Die Rolle der Selektion

Sowohl in der natürlichen Evolution wie auch im Rahmen der algorithmischen Abstraktion ist die Rolle des Selektionsmechanismus nicht zu unterschätzen. Er ist es, der nach Darwin (1859) über die „Bewahrung von vorzuziehenden Varia-

tionen und die Rückweisung schädlicher Variationen” entscheidet<sup>11</sup>. Dabei führt Darwin mehrere Selektionseinflüsse an, etwa die natürliche Umgebung, klimatische Bedingungen oder auch den Einfluss anderer Spezies, die zentral über den Fortpflanzungserfolg (sexuelle Selektion) und damit über das Fortbestehen eines Individuums und in der Konsequenz seiner genetischen Informationen bestimmen. Insofern ist die Selektion ein stabilisierendes Element im Evolutionsprozess und ermöglicht erst einen gerichteten Fortschritt. Dadurch folgt die Evolution – im Gegensatz zu einer Zufallsuche – einem Entwicklungspfad und wird so zielgerichtet. Algorithmisch gesehen werden Lösungseigenschaften aufeinander aufbauend und entsprechend der in Kapitel 1 gegebenen Iterationsvorschrift 1.6 über mehrere Generationen hinweg entwickelt.

Die algorithmische Umsetzung der Selektion verwandelt den in der Natur durch Überlebens- und Fortpflanzungsraten geprägten Mechanismus in eine aktive und zielgerichtete Komponente (Bäck, 1996). Diese gerichtete Ausprägung ist es auch, die allen Selektionsmechanismen gemein ist. Sie besagt, dass mit der Verschiebung von Umweltbedingungen auch eine Verschiebung in der Überlebenswahrscheinlichkeit für an die Bedingungen besser angepasste Phänotypen einhergeht (Campbell & Reece, 2006). Eine weitere Annahme, die allen Mechanismen zugrunde liegt, ist das Vorhandensein einer Ordnung der Individuen, die aufgrund ihrer phänotypischen Ausprägung und der sich daraus ergebenden Fitness hergestellt werden kann.

Im Folgenden soll nun auf drei verschiedene Arten der Selektion eingegangen werden. Dabei dient im Sinne des allgemeinen evolutionären Vorgehens jeder Mechanismus der Auswahl einer Menge von Individuen, der in der nächsten Generation die Fortpflanzung erlaubt wird.

**$(\mu + \lambda)$  und  $(\mu, \lambda)$ -Selektion:** Die Grundzüge der beiden Selektionsarten wurden bereits einleitend erwähnt. Beide Varianten sind vollständig deterministische Methoden. Die Überlebenswahrscheinlichkeit eines Individuums hängt direkt von seiner Fitness innerhalb der Population ab. Dabei stellt die sogenannte Plus-Selektion einen elitären Ansatz dar, der eine monotone Verbesserung der Qualität innerhalb der Population sicherstellt (Bäck, 1996). Die sogenannte Komma-Selektion gibt den Anspruch auf monotone Verbesserung auf und selektiert die neue Elterngeneration aus der Nachkommenpopulation. Beide Ansätze stammen aus dem Bereich der ES und sind dort auch aufgrund ihres deterministischen Vorgehens grundlegend für viele theoretische Analysen (Beyer, 2001) und die Entwicklung von selbstadaptiven Mutationen wie dem Selbstanpassungsmechanismus nach Schwefel (1995).

**Fitnessproportionale Selektion:** Angelehnt an Selektionsregime sexueller Reproduktion innerhalb von Populationen, erlauben die fitnessproportionalen Ansätze grundsätzlich jedem Individuum der Population die Reproduktion,

<sup>11</sup>Engl. orig.: *This preservation of favourable variations and the rejection of injurious variations, I call Natural Selection.* Aus *The Origin of Species* (Darwin, 1859), Kap. 4, S. 51.

jedoch mit unterschiedlicher Wahrscheinlichkeit. Aus diesem Grund gehören diese Ansätze zu den statistischen und dynamischen Methoden (Bäck, 1996). DeJong schlug 1975 eine proportionale Methode vor, die die Reproduktionswahrscheinlichkeit abhängig von der individuellen und der Gesamtfitness der Population bestimmt. Die Reproduktionswahrscheinlichkeit  $p_i$  für das  $i$ -te Individuum  $\vec{x}_i$  ergibt sich dann mit der Fitness  $f(\vec{x}_i)$  nach Gleichung 2.7. Es ist zu beachten, dass dieser Mechanismus insbesondere für Maximierungsprobleme definiert ist<sup>12</sup>. Oftmals wird diese Methode auch als Roulettrad-Methode bezeichnet. Die Analogie verschieden großer Fächer für die Aufnahme der Kugel an einem Roulettrad drückt die unterschiedlichen Wahrscheinlichkeiten für die Wahl eines Individuums aus und spielt insbesondere bei der praktischen Umsetzung des Mechanismus eine Rolle.

$$p_i = \frac{f(\vec{x}_i)}{\sum_{j=1}^N f(\vec{x}_j)}, \text{ mit } \forall j : f(\vec{x}_j) \geq 0 \text{ und } N \text{ Populationsgröße} \quad (2.7)$$

Eine Methode mit ebenfalls relativer Bewertung von Individuen zu der Gesamtpopulation – aber ohne direkte Verwendung der Fitnesswerte der Individuen – schlug Baker (1985) im Rahmen von GAs mit dem sogenannten Ranking vor. Dort werden den Individuen Ränge bezüglich ihrer Qualität zugewiesen, deren Ordnung dann die Wahrscheinlichkeit der Reproduktion bestimmt. Damit ist der Einsatz in Minimierungsproblemen ebenfalls direkt möglich. Baker selbst schlug in seiner Arbeit ein lineares Ranking vor, das die Indizes der Ränge auf entsprechende Fortpflanzungswahrscheinlichkeiten abbildet. Eine Beschränkung auf ein lineares Ranking ist nicht nötig: Michalewicz (1999) etwa nutzt alternativ ein exponentielles Ranking.

**Turnierselektion:** Wendet man eine spezielle Variante des Rankings auf  $k$  (zufällig) ausgewählten Individuen der Gesamtpopulation an und weist zugleich nur dem besten Individuum in dieser Untermenge die Reproduktionswahrscheinlichkeit  $p_{best} = 1$  zu (allen restlichen Individuen  $p_i = 0$ ), so spricht man von der Turnierselektion (Goldberg & Deb, 1991). Diese Selektion wird so oft wiederholt, bis ein entsprechender Pool von Elternindividuen vorliegt<sup>13</sup>. Dabei können natürlich auch Individuen in den Reproduktions-Pool aufgenommen werden, die zuvor mit einer Wahrscheinlichkeit von 0 versehen wurden. In vielen Anwendungen wird  $k = 2$  gewählt (Michalewicz, 1999).

---

<sup>12</sup>Je größer ein Fitnesswert für ein Individuum ist, desto größer ist die Reproduktionswahrscheinlichkeit des Individuums.

<sup>13</sup>Oftmals wird das gewählte Individuum nicht aus dem Selektionspool entfernt. Dies kann jedoch dazu führen, dass es mehrmals in den Reproduktionspool aufgenommen wird und so etwa ein Crossover zwischen gleichen Individuen eine weitere Kopie der Eltern erzeugt. Dies kann mittelfristig zum Diversitätsverlust führen.

### 2.2.4 Populationsstrukturen

Als letzter Punkt der allgemeinen Betrachtung von Eigenschaften eines evolutionären Algorithmus bleibt noch die Populationsstruktur. Die Biologie betrachtet eine Population folgendermaßen:

*„Eine Population ist eine Gruppe von Individuen einer Art, die einen gemeinsamen Lebensraum bewohnen. Ihre Mitglieder sind damit auf dieselben Ressourcen angewiesen, ähnlichen Umfeldfaktoren ausgesetzt und paaren und interagieren mit großer Wahrscheinlichkeit miteinander.“* (aus Campbell & Reece (2006), Kap. 52, S. 1376)

Bei dieser Betrachtung sind zwei Aspekte für die Übertragung in Populationsmodelle für die evolutionären Algorithmen wichtig, nämlich der gemeinsame Lebensraum und die Paarungs- und Interaktionswahrscheinlichkeit. Beide Aspekte implizieren einen Spielraum, der ganz unterschiedliche Ausprägungen einer Populationsstruktur erlaubt. Die grundlegend angenommene Populationsstruktur ist panmiktisch und erlaubt die Paarung zweier beliebiger Individuen mit gleicher Wahrscheinlichkeit über die gesamte Population, also eine quasi beschränkungsfreie Fortpflanzung. Beschränkungen im Lebensraum können aber leicht zu einer individuellen Veränderung der Paarungswahrscheinlichkeiten führen. Diese Dispersion<sup>14</sup> der Population kann unterschiedliche Ursachen haben (Leben in Herden, Umwelteinflüsse) und zu einer Abgrenzung von Populationsteilen mit geringerer genetischer Wechselwirkung untereinander führen.

Im Rahmen der evolutionären Algorithmen kann man drei grobe Unterscheidungen in der Populationsstruktur identifizieren, die in einer Spannweite von unstrukturiert bis stark strukturiert zu bezeichnen sind (Alba & Dorronsoro, 2008): panmiktische Populationen, verteilte Populationen und zellulare Populationen<sup>15</sup>. Alle drei Varianten werden nun kurz dargestellt.

**Panmiktische Population:** Die panmiktische Darstellung entspricht derjenigen biologischen Population, in welcher sich jedes Individuum mit jedem anderen Individuum populationsübergreifend und mit gleicher Wahrscheinlichkeit paaren kann. Es gibt also keine Hürden für die Interaktion von Individuen, was zu einer vollständig unstrukturierten Population führt (Tomassini, 2005). Abbildung 2.3(A) stellt die panmiktische Population dar. Sie ist die am weitesten verbreitete Variante, die in evolutionären Algorithmen Anwendung findet. Der Selektionsmechanismus ist auf dieser Population global definiert, bezieht also die gesamte Population in die Bewertung ein. Bei der Reproduktion hingegen kann unterschieden werden, ob die gesamte Population der Variation unterliegt oder nur einige Elemente. Der zur globalen oder generationenbezogenen Variation komplementäre Fall ist durch die Reproduktion

<sup>14</sup>Räumliche Verteilung der Population.

<sup>15</sup>Engl.: Cellular population; Die Bezeichnung ergibt sich aus der feingranularen Aufteilung der Population in einzelne Zellen, den Zellarraum.

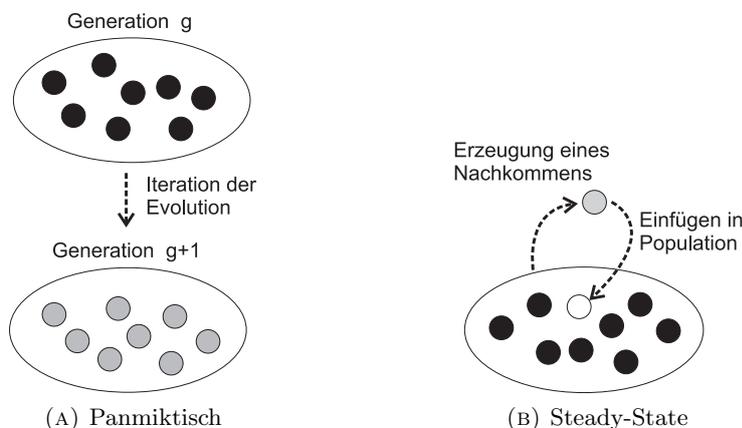


ABBILDUNG 2.3: Darstellung der panmiktischen Population und des Spezialfalles der Steady-State-Variante.

eines einzigen Individuums gegeben. In diesem Fall spricht man von einem „Steady-State“-Ansatz, da die Population weitgehend in einem stabilen Status verweilt und nur ein einziges Individuum ersetzt wird. Schematisch ist dieser Fall in Abbildung 2.3(B) dargestellt.

**Verteilte Population:** Im Gegensatz zur Panmixie erlaubt das Inselmodell als einfachste Art der Verteilung unterschiedliche Reproduktionswahrscheinlichkeiten zwischen Individuen gleicher Spezies. Entsprechend der natürlichen Herdenbildung werden Teilpopulationen gebildet die in sich wieder Panmixie erlauben, aber voneinander weitgehend getrennt existieren. Einzig eine Migration zwischen den Teilpopulationen erlaubt einen Austausch von Geninformationen. Abbildung 2.4(A) zeigt dies beispielhaft für drei Teilpopulationen mit Migration zwischen den Teilpopulationen.

**Zellulare Population:** Die stärkste Strukturierung ist mit zellularen Populationen zu erreichen, die auf dem Konzept kleiner Nachbarschaften basieren.<sup>16</sup> Dabei wird die Selektion und Reproduktion insbesondere aus Sicht eines einzelnen Individuums betrachtet, das sich in der Regel mit direkt benachbarten Individuen paart und nur mit geringer Wahrscheinlichkeit oder überhaupt nicht auf weit entfernte Individuen trifft (Alba & Dorronsoro, 2008). Die Einführung einer Entfernung impliziert direkt die starke Strukturierung der Population. Häufig wird dazu auf eine Anordnung im euklidischen Raum oder auf Graphenstrukturen zurückgegriffen, siehe Abbildung 2.4(B).

Die Strukturierung von Populationen ist Grundlage für verschiedene Ansätze zur Parallelisierung von Algorithmen und zugleich für die Hoffnung auf eine bessere Explorationsfähigkeit des Algorithmus. Bezüglich der Parallelisierung kann ei-

<sup>16</sup>Baluja verwendet diese Art der Strukturierung im Zusammenhang mit sogenannten feingraularen parallelen GAs (engl. fine-grained parallel GAs), siehe Baluja (1993).

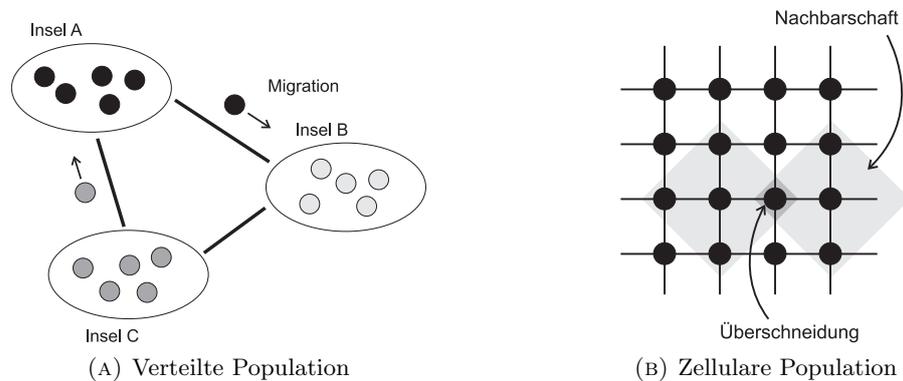


ABBILDUNG 2.4: Darstellung der Varianten von verteilten Populationen.

ne stärkere Strukturierung genutzt werden, um algorithmische Komponenten auf verschiedene Prozessoren oder als nebenläufige Prozesse zu verteilen. Bei der panmiktischen Population bietet sich nur eine einzige Möglichkeit der Parallelisierung des evolutionären Algorithmus: über die Auswertung der Individuen selbst. Jede Auswertung kann von einem anderen Prozessor durchgeführt werden, so dass sehr zeitintensive Berechnungen nebenläufig und damit effizienter als in einem sequentiellen Szenario durchgeführt werden können (Luque, Alba & Dorronsoro, 2005). Dabei ist zu beachten, dass der generationenbasierte Selektionsmechanismus einen natürlichen Synchronisationspunkt für die Prozesse darstellt. Um also eine gleichmäßige Verteilung der durch die Auswertungen verursachten Last zu erreichen, müssen entsprechende Planungs- und Verteilungsmechanismen in den Algorithmenentwurf einbezogen werden. Selbst mit solchen Verfahren kann eine einzige Auswertung die Ausführungszeit aller anderen Auswertungen stark dominieren und durch die Generationenschranke zu starken Verzögerungen führen (Cantú-Paz, 2005). Insgesamt ändert sich, bis auf einen eventuellen Geschwindigkeitsgewinn, nichts an dem Lösungsverhalten des Algorithmus verglichen mit der sequentiellen Ausführung. Der stärker strukturierte Ansatz des Inselmodells erlaubt es dagegen mehrere kleine Algorithmeninstanzen parallel laufen zu lassen. Im Extremfall, wenn keine Migration von Lösungen zwischen den Populationen erlaubt wird, kann man von einer parallel ausgeführten Multistartvariante des Algorithmus sprechen. Der eigentliche Algorithmus wird auf mehreren Teilpopulationen unabhängig voneinander ausgeführt. Da evolutionäre Algorithmen randomisierte Suchverfahren sind, kann eine solche Aufteilung zu einem stärkeren Explorationsverhalten führen als die Anwendung des Algorithmus auf eine einzige – wenn auch größere – Population. Trotzdem sind die Individuen zwischen den Teilpopulationen vergleichbar, so dass aus unterschiedlichen Lösungen die Beste gewählt werden kann. Lässt man zusätzlich eine Migration von Individuen aus einer Teilpopulation in eine andere Teilpopulation zu, so kann sich ein eventueller Einfluss aus einer Population in eine andere fortpflanzen und dort zu einer Beschleunigung der Lösungsfindung führen. Andererseits ist dieser Parameter mit Bedacht

zu wählen, da ein zu starkes Migrationsverhalten die Unabhängigkeit der Populationsentwicklung und damit die diversitätserhaltende Funktion stark einschränken kann (Baluja, 1993). Durch eine noch stärkere Strukturierung im Sinne der zellularen Population kann leicht eine Individuen-bezogene Sicht auf die Population eingenommen werden. Jedes Individuum existiert in einer sehr kleinen, abgeschlossenen Umgebung, die an die Umgebungen anderer Individuen grenzt. So kann jedes Individuum einen einzelnen Prozessor einnehmen und nebenläufig mit anderen Individuen agieren. Der Interaktionsradius beschränkt sich dabei gewöhnlich auf eine lokale Nachbarschaft. Dies verringert einerseits die Granularität der Parallelisierung des Algorithmus, verlagert zudem die Selektion noch stärker auf eine lokale Ebene als bereits das Inselmodell, und stellt dadurch ebenfalls eine gewisse Diversitätserhaltung sicher. Genetische Änderungen eines Individuums setzen sich andererseits durch die ausschließlich lokale Interaktion eher langsam in der gesamten Population durch. Diese Populationsart und die Auswirkungen der Lokalität werden in Kapitel 5 dieser Arbeit sowohl unter den Aspekten der Parallelisierung wie auch der Lösungsentwicklung genauer betrachtet.



## Kapitel 3

# Algorithmen für mehrkriterielle Problemstellungen

Nachdem in den vorangegangenen Kapiteln ausführlich die Grundlagen der Optimierung und evolutionäre Verfahren als allgemeine randomisierte Lösungsstrategien betrachtet wurden, wird an dieser Stelle auf die Lösung mehrkriterieller Problemstellungen im Speziellen eingegangen. Dabei geht es zunächst um grundlegende, deterministische Ansätze, die im Laufe der Arbeit noch verstärkt im Zusammenspiel mit dem hier weiterentwickelten und untersuchten evolutionären Ansatz betrachtet werden. Zudem werden anschließend einige etablierte aber auch unkonventionelle evolutionäre Verfahren betrachtet, die einerseits die Bandbreite der Methodik darstellen sollen und andererseits ein Ausgangspunkt für die Einordnung des in der Arbeit betrachteten Ansatzes sind.

Einleitend sei noch kurz auf eine oftmals in der Literatur getroffene Unterscheidung von sogenannten *a-posteriori*, *a-priori* und *interaktiven* Methoden zur mehrkriteriellen Optimierung hingewiesen (Coello et al., 2007; Miettinen, 1999). Diese Klassifikationen unterscheiden insbesondere nach dem in die Lösungsbildung einfließenden Wissen und der Menge der gebildeten Lösungen. Erstere Methoden streben an, die gesamte Paretofront zu erzeugen und damit einem Entscheidungsträger nach Anwendung des Optimierungsverfahrens die Auswahl aus der Menge ermittelter Lösungen zu überlassen. Dies setzt keine Information über die Beschaffenheit des Lösungsraumes voraus. A-priori-Methoden hingegen verlangen eine vorherige Festlegung des Entscheidungsträgers auf gewisse Rahmenbedingungen für die Optimierung, etwa eine Gewichtung der verfolgten Ziele, oder eine Definition von Grenzen für Funktionswerte einzelner Kriterien. Dies setzt aber die Kenntnis über den Problem- oder Lösungsraum voraus. Die interaktive Methode erlaubt den Eingriff des Entscheidungsträgers in den Optimierungsprozess, um während der Exploration eventuelle Anpassungen vorzunehmen und die Suche durch Parameteranpassung oder Suchraumbegrenzung in gewünschte Regionen zu lenken. Wir gehen im Folgenden generell von dem allgemeinen Fall der a-posteriori-Methoden

aus, da Problemstellungen ohne Lösungsinformationen das Hauptanwendungsgebiet evolutionärer Verfahren sind.

### 3.1 Deterministische Ansätze zur mehrkriteriellen Optimierung

Die hier diskutierten Verfahren sind für die Ermittlung einer Näherung der Paretofront prinzipiell zweistufig zu betrachten. Das mehrkriterielle Optimierungsproblem *MOP* aus Definition 1.6 wird dabei zumeist in ein spezielles einkriterielles Problem umformuliert und anschließend durch ein geeignetes Suchverfahren gelöst. Dadurch ergibt sich *eine* Lösung für *MOP*. Anschließend wird die Formulierung der einkriteriellen Instanz geringfügig geändert, was zu einer weiteren Lösung führt. So ergibt sich iterativ eine Menge an Lösungen, die auf oder nahe der Paretofront liegen.

#### 3.1.1 Methode der Gewichtung

Eine weit verbreitete und einfache Umformulierung eines *MOP* ergibt sich durch die Aggregation der einzelnen Kriterien mittels Gewichtungen. Dabei stellt  $\vec{w} = (w_1, \dots, w_m)^T$ ,  $0 \leq w_i \in \mathbb{R}$  den Gewichtsvektor für alle  $m$  Kriterien dar. Über die Gewichtungen kann jedem einzelnen Kriterium eine Priorität zugeordnet werden. Gewöhnlicherweise wird das Problem mittels normierter Gewichtungen formuliert:

$$\overline{MOP}^w : \vec{w}f = \sum_{i=1}^m w_i f_i(\vec{x}) \rightarrow \min \quad (3.1)$$

$$\text{wobei } \vec{x} \in S \text{ und } \sum_{i=1}^m w_i = 1$$

Durch die entsprechende Wahl eines Gewichtsvektors und die Anwendung einer einkriteriellen Lösungsstrategie kann eine Lösung auf der Paretofront bestimmt werden. Dabei ist zu beachten, dass diese Methode nur bei konvexen Paretofronten uneingeschränkt einsetzbar ist, die generierten Lösungen aber nicht notwendigerweise eindeutig sind (Miettinen, 1999). Nicht konvexe Bereiche der Paretofront können mit der Methode nicht gefunden werden. Dies zeigen Das & Dennis (1997) anschaulich anhand einer trigonometrischen Formulierung von Gleichung 3.1. Eine knappe Darstellung der Beweisidee findet sich im Anhang A.1 der Arbeit.

Abschließend soll hier noch erwähnt werden, dass eine einzelne Iteration der Gewichtungsmethode natürlich als a-priori-Ansatz gewählt werden kann. Der Entscheidungsträger muss dann initial die Gewichtungen der Einzelkriterien wählen.

#### 3.1.2 Epsilon-Constraint-Methode

Die von Haimes, Lasdon & Wismer (1971) vorgeschlagene  $\epsilon$ -Constraint-Methode wandelt – wie zuvor die Gewichtungsmethode – ein mehrkriterielles Problem in ein

einkriterielles Problem um. Dabei wird jedoch ein einzelnes Kriterium  $f_c(\vec{x})$ ,  $c \in \{1, \dots, m\}$  als Zielfunktion gewählt, während für alle übrigen Kriterien obere Grenzen eingeführt werden. Diese sogenannten  $\epsilon$ -Randbedingungen  $\vec{\epsilon} = (\epsilon_1, \dots, \epsilon_m)$  und  $\epsilon_c = \infty$  legen fest, welcher Wert  $\epsilon_i$  für ein Kriterium  $f_i$  nicht überschritten werden darf. Es ergibt sich formal das in Vorschrift 3.2 gegebene Minimierungsproblem.

$$\begin{aligned} \overline{MOP}^\epsilon : f_c(\vec{x}) \rightarrow \min & \quad (3.2) \\ \text{wobei } \vec{x} \in S \text{ und } \forall i = 1, \dots, m, i \neq c : f_i(\vec{x}) \leq \epsilon_i & \end{aligned}$$

Wie auch bei der vorherigen Methode muss diese Minimierung mit unterschiedlichen Randbedingungen iterativ ausgeführt werden, um eine gute Abdeckung der Paretofront zu erhalten. Im Gegensatz zur Gewichtungsmethode ist der  $\epsilon$ -Constraint-Ansatz in der Lage, auch Lösungen in nicht konvexen Bereichen der Paretofront zu finden. Diese Eigenschaft ergibt sich direkt aus der Feststellung, dass ein Vektor  $\vec{x} \in S$  nur genau dann Pareto-optimal ist, wenn er eine Lösung der  $\epsilon$ -Constraint-Methode ist.

**Beweis:** Wir benutzen die Definition aus Vorschrift 3.2 und nehmen an, dass  $\hat{x} \in S$  optimal ist, aber keine Lösung der  $\epsilon$ -Constraint-Methode mit einem  $c$  und  $\epsilon_i = f_i(\hat{x})$  für  $i = 1, \dots, m$ ,  $i \neq c$ . Dann sollten wir ein  $x' \in S$  mit  $f_c(x') < f_c(\hat{x})$  und  $f_i(x') \leq f_i(\hat{x})$  finden. Dies würde aber gerade die Optimalität von  $\hat{x}$  verletzen.

Wenn wir weiterhin annehmen, dass  $\hat{x}$  eine Lösung der  $\epsilon$ -Constraint-Methode ist, dann gibt es gemäß Vorschrift 3.2 kein  $x' \in S$  mit  $f_c(x') < f_c(\hat{x})$  und  $f_i(x') \leq f_i(\hat{x})$ . Dies ist aber gerade die Definition der Pareto-Optimalität.

□

Der kurze Beweis ist bei Miettinen (1999) entlehnt. Die dadurch gezeigte breitere Anwendbarkeit der Methode wird allerdings durch einen größeren Aufwand zur Ermittlung der Lösungen erkauft. Gleichzeitig muss diese Methode mit sinnvollen Randbedingungen initialisiert werden, sie setzt also eine gewisse Kenntnis des Lösungsraums voraus (Miettinen, 1999). Wie bei der vorherigen Methode kann eine einzelne Iteration der  $\epsilon$ -Constraint-Methode als a-priori-Ansatz verwendet werden, wenn der Entscheidungsträger ein primäres Kriterium und geeignete  $\epsilon$ -Randbedingungen wählt.

### 3.1.3 Tchebycheff-, Zenitpunkt- oder Compromise Programming-Methode

Dieser mit vielen Bezeichnungen versehene und von Bowman (1976) mit Bezug zur Tchebycheff-Norm vorgeschlagene Ansatz basiert ebenfalls auf der Anwendung von Gewichtungen, betrachtet bei der Minimierung in jeder Iteration aber nur jenes Kriterium, das bezüglich des Funktionswertes den größten Abstand zu einem Referenzpunkt aufweist. Als Referenzpunkt  $\vec{u}^*$  wird gewöhnlich der sogenannte

Utopia,- Ideal- oder Zenitpunkt<sup>1</sup> gewählt. Dieser stellt einen virtuellen Zielpunkt im Bildraum dar, und besteht in seinen Komponenten aus den optimalen Werten der Einzelkriterien (Miettinen, 1999). Die Umformulierung des mehrkriteriellen Problems geschieht in diesem Falle über die gewichtete Tchebycheff-Norm<sup>2</sup>. Vorschrift 3.3 stellt das Minimierungsproblem für einen Gewichtungsvektor  $\vec{w}$ , einen Utopiapunkt  $\vec{u}^*$  und  $m$  Kriterien formal dar.

$$\overline{MOP}^t : \max_{i=1,\dots,m} \{w_i \cdot \| f_i(\vec{x}) - \vec{u}_i^* \| \} \rightarrow \min \quad (3.3)$$

wobei  $\vec{x} \in S$

Diese Methode hat ebenfalls den Vorteil, dass sie Lösungen im nicht-konvexen Bereich der Paretofront findet.

**Beweis:** Wir zeigen entsprechend des Vorgehens in Miettinen (1999): Sei  $\vec{x}^* \in S$  Pareto-optimal. Dann gibt es einen Gewichtungsvektor  $\vec{0} < \vec{w} \in \mathbb{R}^n$  so dass  $\vec{x}^*$  eine Lösung der Tchebycheff-Methode mit dem Referenzpunkt  $\vec{u}^*$  ist.

Wir nehmen an, dass kein solcher Gewichtungsvektor  $\vec{w}$  existiert, der  $\vec{x}^*$  als Lösung der Tchebycheff-Methode erzeugt. Da  $\vec{u}^*$  unerreichbar ist, wissen wir, dass  $f_i(\vec{x}) > \vec{u}_i^*$  gilt. Wir wählen zusätzlich einen Faktor  $\gamma > 0$ , so dass  $w_i = \frac{\gamma}{f_i(\vec{x}^*) - \vec{u}_i^*}$  für alle  $i = 1, \dots, m$ . Wenn unsere Gegenannahme gilt, dann muss es ein  $\vec{x}'$  geben so dass gilt:

$$\begin{aligned} \max_{i=1,\dots,m} \{w_i \cdot f_i(\vec{x}') - \vec{u}_i^*\} &< \max_{i=1,\dots,m} \{w_i \cdot f_i(\vec{x}^*) - \vec{u}_i^*\} \\ &= \max_{i=1,\dots,m} \left\{ \frac{\gamma}{f_i(\vec{x}^*) - \vec{u}_i^*} (f_i(\vec{x}') - \vec{u}_i^*) \right\} \\ &= \gamma \end{aligned}$$

Dies ergibt direkt:

$$\begin{aligned} \frac{\gamma}{f_i(\vec{x}^*) - \vec{u}_i^*} (f_i(\vec{x}') - \vec{u}_i^*) &< \gamma \\ \Leftrightarrow f_i(\vec{x}') &< f_i(\vec{x}^*) \end{aligned}$$

Dies steht aber im Widerspruch zur Optimalitätsannahme von  $\vec{x}^*$ . □

Als Problem der Methode kann man bezeichnen, dass neben Pareto-optimalen Lösungen ebenfalls schwach Pareto-optimale Lösungen erzeugt werden können. Diese müssen mit zusätzlichem Aufwand aus der Lösungsmenge entfernt werden. Wie bei allen bisher vorgestellten Methoden, deren iterative Anwendung (hier mit unterschiedlichen Gewichtungsvektoren) erst zu einer guten Abdeckung der Paretofront führen, kann auch dieser Ansatz als a-propri-Methode eingesetzt werden.

<sup>1</sup>Die drei Bezeichnungen werden in der Literatur synonym benutzt.

<sup>2</sup>Diese Norm wird in der Literatur oftmals mit mehreren Namen belegt (Tchebycheff-Norm, Supremumsnorm oder auch Tchebycheff-Distanz), wenn sie sich auf den euklidischen Raum bezieht (Storch & Wiebe, 1999).

## 3.2 Evolutionäre Verfahren für die mehrkriterielle Optimierung

Die zuvor aufgeführten deterministischen mehrkriteriellen Optimierungsmethoden basieren bei genauer Betrachtung auf einem iterativen Schema, das Einzellösungen durch die wiederholte Anwendung einkriterieller Verfahren ermittelt. Eine Paretofront entsteht – wenn überhaupt – erst durch die vielmalige Anwendung mit unterschiedlicher Parametrisierung.

Aus der Betrachtung der evolutionären Algorithmen in Kapitel 2 ergibt sich ein ebenfalls iterativer aber mengenbasierter Ansatz: die Verwendung einer Population impliziert bereits die gleichzeitige Generierung verschiedener Lösungen. Während die Vielfältigkeit der Lösungen jedoch bei einkriteriellen Problemstellungen gewöhnlicherweise nahe der global optimalen Lösung abnimmt, ist die Lösungsdiversität bei mehrkriteriellen Problemstellungen neben der Annäherung an die Lösungsmenge ein gleichwertiges, zweites Ziel. Somit liegt die Übertragung von evolutionären Ansätzen auf mehrkriterielle Problemstellungen aufgrund der Erzeugung von Lösungsmengen nahe, verlangt aber, dass die Diversität gleichwertig zur Konvergenz betrachtet wird.

Bei der Anwendung von evolutionären Algorithmen auf mehrkriterielle Optimierungsprobleme ergeben sich im Unterschied zur einkriteriellen Problemstellung zwei Schwierigkeiten:

**Bewertung von Lösungen:** Wie bereits in Definition 1.1 dargestellt, ist die Dominanzrelation  $\preceq$  für mehrkriterielle Problemstellungen nur als partielle Ordnung gegeben. Sie kann also nicht uneingeschränkt im Sinne der Selektion gewöhnlicher evolutionärer Algorithmen zum Vergleich zweier Lösungen verwendet werden, da viele Lösungen unvergleichbar sind.

**Diversitätserhaltung:** Da die Paretofront aus mehreren nicht vergleichbaren Lösungen besteht, müssen diese bei einer einzigen Anwendung des Algorithmus erreicht werden und erhalten bleiben, um eine möglichst vollständige Abdeckung der Lösungsmenge zu erreichen. Jedoch steht im Kontext der Effizienz eines evolutionären Algorithmus die Unterschiedlichkeit der Individuen und damit die Explorationseigenschaft im Gegensatz zur schnellen Annäherung an die Paretofront (Bosman & Thierens, 2003).

Da in den meisten Verfahren die grundlegende Evolutionsschleife aus Reproduktion, Variation und Selektion genutzt wird, müssen die angesprochenen Probleme der Bewertung und Diversitätserhaltung vordringlich bearbeitet werden. Trotz dieser generellen Einsicht, haben sich zwei große Klassen unterschiedlicher Ansätze entwickelt, die die obigen Probleme im Rahmen eines komplexen Selektionsmechanismus angehen: dominanzbasierte Methoden und indikatorbasierte Ansätze. Aus beiden Klassen werden in diesem Kapitel wichtige Vertreter diskutiert. Daneben gibt es eine Vielzahl sehr unterschiedlicher alternativer Ansätze, die ebenfalls in

diesem Kapitel benannt werden und damit den Rahmen für die Einordnung des hier untersuchten Ansatzes darstellen.

Interessanterweise beginnt die Übertragung von evolutionären Verfahren auf mehrkriterielle Problemstellungen in der Literatur nicht mit dominanz- oder indikatorbasierten Ansätzen, sondern mit einem Ansatz von Schaffer (1985), der Teilpopulationen ausschließlich einkriteriell betrachtet. Zwar behält Schaffer in seinem Vector Evaluated Genetic Algorithm (VEGA) eine Generationenfolge bei, zerlegt aber die Selektion in Teilaktivitäten: Der Ansatz wählt  $k \leq m$  ( $m$  Anzahl der Kriterien) Teilmengen der Gesamtpopulation bezüglich eines jeweils einzigen Kriteriums. In der resultierenden Population wird die Trennung in Teilmengen wieder aufgehoben und Variationsoperatoren angewendet, um Nachkommen zu erzeugen (Schaffer & Grefenstette, 1985). Insofern unterscheidet sich VEGA nur aufgrund des Selektionsmechanismus von einem gewöhnlichen GA.

### 3.2.1 Dominanzbasierte Methoden

Mit Goldbergs Vorschlag die Pareto-Dominanz als Kriterium für das Erstellen einer Rangfolge von Lösungen zu nutzen (engl.: Ranking) beginnt die Entwicklung von tatsächlich mehrkriteriellen Selektionsmechanismen für evolutionäre Verfahren (Goldberg, 1989). Diese Mechanismen basieren auf dem einfachen Prinzip Lösungen entsprechend der Dominanzrelation einen Rang zuzuweisen. Beispielhaft sollen zwei Methoden dargestellt werden, siehe auch Abbildung 3.1.

Betrachtet man jede Lösung individuell im Vergleich zur Lösungsmenge, so kann ein Rang festgelegt werden, indem die Anzahl derjenigen Lösungen bestimmt wird, die die individuelle Lösung dominieren. Dieser Wert wird um 1 erhöht und als Rang zugewiesen.<sup>3</sup> Die im Folgenden als Individualranking bezeichnete Methode ist beispielhaft in Abbildung 3.1(A) dargestellt. Eine Alternative bietet die Bestimmung aller nicht dominierten Lösungen. Dieser Menge wird der Rang 1 zugewiesen und anschließend werden die entsprechenden Individuen aus der Lösungsmenge entfernt. Auf der übrigen Lösungsmenge wird der obige Vorgang der Rangbildung wiederholt und den nicht dominierten Lösungen der nächsthöhere Rang zugeordnet. Dies wird fortgesetzt, bis alle Individuen einem Rang bzw. einer Teilfront zugeordnet sind. Das sogenannte Frontranking ist in Abbildung 3.1(B) dargestellt. Mit beiden Methoden entsteht eine grobe Unterteilung der Lösungsmengenqualität bezüglich der Dominanzrelation. Für eine feinere Unterscheidung innerhalb der Menge eines Ranges kann zusätzlich auf die Verteilung der Lösungen zurückgegriffen werden.

### Algorithmische Varianten

Die dominanzbasierten Algorithmen nutzen in ihrem Selektionsmechanismus die vorgestellte Rangbildung als grundlegendes Prinzip. Zusätzlich wird oftmals ein

---

<sup>3</sup>So erhalten nicht dominierte Lösungen den Rang 1.

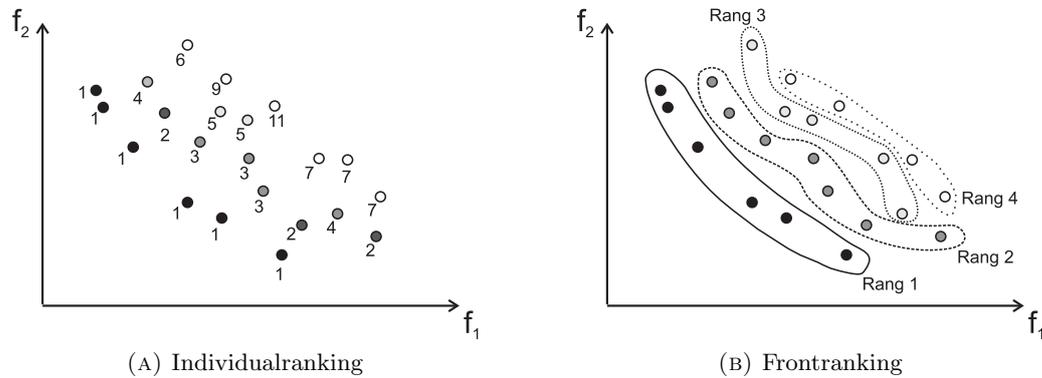


ABBILDUNG 3.1: *Beispielhafte Darstellung zweier unterschiedlicher Ranking-Mechanismen. Links entsteht der Rang aus der Anzahl dominierender Lösungen bzgl. aller Kriterien plus 1. Rechts wird der Rang durch die Zugehörigkeit zur entsprechenden nicht dominierten Teilmenge bestimmt.*

Mechanismus eingesetzt, der unvergleichbare Lösungen entsprechend ihrer Verteilung bewertet. Als erste Beispiele solcher Algorithmen seien hier der Multi-Objective Genetic Algorithm (MOGA) von Fonseca & Fleming (1993) sowie der Niche-Pareto Genetic Algorithm (NPGA) von Horn, Nafpliotis & Goldberg (1994) erwähnt. Beide verwenden das Individualranking und jeweils eine Variante des sogenannten phänotypischen Fitness-Sharings<sup>4</sup> zur Erhaltung der Diversität der Population (Deb & Goldberg, 1989), unterscheiden sich aber stark im Selektionsmechanismus. MOGA ordnet zunächst die gesamte Population mit einem entsprechenden Individualranking, und präzisiert anschließend die individuelle Fitness einer Lösung mit der Lösungsdichte der Nachbarschaft. Im Gegensatz dazu erstellt NPGA eine neue Population iterativ durch die zufällige Auswahl zweier Individuen und die Anwendung eines sogenannten *Pareto domination tournaments* (Horn et al., 1994). Dabei werden die zwei zuvor gewählten Kandidatenlösungen mit einer Untergruppe der Population verglichen (nach Coello Coello et al. gewöhnlich 10 % der Populationsgröße (Coello et al., 2007)). Entscheidend für die Aufnahme in die Folgepopulation ist: Wird einer der Kandidaten von der Teilpopulation dominiert und der andere Kandidat nicht, so wird die nicht-dominierte Lösung übernommen. Werden hingegen beide Kandidaten dominiert oder dominieren sie beide die

<sup>4</sup>Das Fitness-Sharing-Schema drückt allgemein die Ähnlichkeit zweier Lösungen in einem Lösungsraum aus. Im phänotypischen Raum ist dieses Maß gewöhnlich über die euklidische Distanz  $d$  und die Funktion

$$S(d) = \begin{cases} 1 - \left(\frac{d}{\sigma_{share}}\right)^\alpha, & \text{wenn } d < \sigma_{share} \\ 0, & \text{sonst.} \end{cases}$$

gegeben. Dabei ist  $\sigma_{share}$  der Nachbarschaftsradius (engl. niche radius). Die Nachbarschaftsdichte wird dann über die Anzahl und Nähe der Lösungen in einem Nachbarschaftsradius bestimmt und kann zur Veränderung der Fitness des untersuchten Individuums eingesetzt werden (Deb & Goldberg, 1989; Coello et al., 2007), um damit schließlich den Reproduktionserfolg sehr dicht gedrängter Individuen zu verringern und einzeln stehende Lösungen zu bevorzugen.

Teilpopulation, so wird dieser "Gleichstand" durch die Betrachtung der jeweiligen Nachbarschaft der Kandidaten aufgehoben. Derjenige Kandidat mit geringerer Nachbarschaftsdichte, also mit weniger ähnlichen Nachbarn in unmittelbarer Nähe, wird in die nächste Generation übernommen.

Die Pareto Archived Evolutionary Strategy (PAES) von Knowles & Corne (2000) nutzt ebenfalls ein *Pareto domination tournament* zur Selektion, führt aber ein Archiv von bereits ermittelten Lösungen als neues Element ein. Dieses Archiv wird während der Selektion genutzt, um einerseits Lösungen zu konservieren und um andererseits ein Individuum entsprechend des Mechanismus aus dem NPGA zu selektieren. Durch die Trennung von Population und Selektionsmechanismus ermöglichen die Autoren sogar die Konstruktion einer (1 + 1)-Variante des PAES: Initial wird eine zufällige Kandidatenlösung gewählt und dem Archiv hinzugefügt. Anschließend wird diese mit einem durch Mutation erzeugten Nachkommen auf Grundlage der mehrkriteriellen Dominanzrelation verglichen. Dominiert die Kandidatenlösung den Nachkommen, so wird der Nachkomme verworfen. Im umgekehrten Falle wird der Nachkomme zur neuen Kandidatenlösung und zugleich in das Archiv übernommen. Treffen beide vorherigen Bedingungen nicht zu (d.h. Kandidatenlösung und Nachkomme sind unvergleichbar), so muss generell überprüft werden, ob der Nachkomme von einer Lösung im Archiv dominiert wird. Ist dies der Fall, so wird der Nachkomme verworfen. Ansonsten muss entschieden werden, ob die Nachkommenlösung in das Archiv übernommen wird, und welche der beiden Lösungen (Nachkomme oder Kandidatenlösung) die neue Kandidatenlösung wird. Dieser Prozess – ausgenommen der initiale Erzeugungsschritt – wird fortgeführt, bis ein Terminierungskriterium erreicht ist. Jede Aufnahme eines Individuums in das Archiv und insbesondere der letzte Entscheidungsschritt gehen wieder mit einer Beurteilung der Diversität einher. Einerseits wird mit der Aufnahmeentscheidung die maximale Größe des Archives sichergestellt. Andererseits löst dieser Mechanismus den möglichen "Gleichstand" in der Lösungsqualität zweier Individuen während der Selektion auf. Knowles und Corne nutzen hier ein sogenanntes Crowding-Maß, das auf der gleichmäßigen Aufteilung des durch die archivierten Individuen überdeckten Lösungsraums basiert, siehe bei Coello et al. (2007) für eine detaillierte Darstellung.

Der Vollständigkeit halber soll an dieser Stelle auf den Strength Pareto Evolutionary Algorithm (SPEA) von Zitzler und Thiele hingewiesen werden (Zitzler & Thiele, 1999; Zitzler, Laumanns & Thiele, 2002). Dieser Ansatz nutzt sein Archiv als ausschließliche Referenz zur Bewertung der eigentlichen Individuen der momentanen Population. Die Fitnesszuweisung wird über sogenannte Stärkewerte (engl.: Strength) durchgeführt, die sich für jedes Individuum  $i$  des Archivs  $A$  durch das Verhältnis  $s_i = \frac{c}{|P|+1}$  ermitteln, wobei  $c$  die Anzahl der in der Population  $P$  von dem Individuum  $i \in A$  dominierten Lösungen ist. Der Fitnesswert eines jeden Individuums  $j \in P$  ergibt sich dann als  $f(j) = 1 + \sum_{i \in A, i \preceq j} s_i$  und wird groß, wenn viele Individuen des Archivs das Individuum  $j$  dominieren. Es sei angemerkt, dass eine Minimierung des Fitnesswertes angestrebt wird. Insgesamt ist die Fitnesszuweisung angelehnt an das Individualranking von MOGA. Zugleich wird

durch die Fitnesszuweisung aber auch das Streben nach Diversität der Lösungen unterstützt. Eine Häufung von Lösungen im Dominanzbereich<sup>5</sup> einer archivierten Lösung führt automatisch zu einem höheren Fitnesswert, also einer schlechteren Bewertung der entsprechenden Lösungen. Zusätzlich ist SPEA mit einem Mechanismus zur Begrenzung der Archivgröße versehen, um dem Entscheidungsträger eine Lösungsmenge von sinnvoller Größe zur Verfügung zu stellen. Diese sogenannte Pruning-Methode stellt sicher, dass das Archiv nicht eine unendliche Größe annimmt. Eine spätere Erweiterung des ursprünglichen Algorithmus zu der SPEA2-Variante verändert insbesondere diesen Mechanismus, so dass Randlösungen der Paretofront erhalten bleiben. Zusätzlich wird in SPEA2 eine detailliertere Fitnesszuweisung eingeführt, die für ein Individuum berücksichtigt, wieviele andere Lösungen es einerseits dominiert und durch welche Anzahl von Lösungen es selbst dominiert wird. Somit wird zur Bewertung und bei der Selektion nun auch die Population und nicht ausschließlich das Archiv herangezogen. Schließlich wird wie in vorherigen Ansätzen ebenfalls die Nachbarschaftsdichte für die Auflösung von "Gleichständen" bei der Fitnessbewertung verwendet.

### **NSGA und NSGA-2**

An dieser Stelle soll der Non-dominated Sorting Genetic Algorithm (NSGA) in seinen zwei Varianten dargestellt werden (Srinivas & Deb, 1994; Deb, Pratap, Agarwal & Meyarivan, 2002). Das Grundprinzip des Algorithmus unterscheidet sich nicht von den zuvor diskutierten algorithmischen Varianten der dominanzbasierten Ansätze. Er nimmt in der Literatur zur evolutionären mehrkriteriellen Optimierung eine prominentere Position ein und wird inzwischen als Standardalgorithmus angesehen. Seine starke Orientierung an der Evolutionsschleife und der daraus resultierende einfache Aufbau des Algorithmus sind sicherlich ebenfalls Gründe für die verbreitete industrielle Anwendung dieses Algorithmus. Dies und die Verfügbarkeit vielfältiger und gut getesteter Implementierungen des Ansatzes machen daraus ein geeignetes Verfahren zu Vergleichszwecken.

Wie bereits dargestellt, ist der Ausgangspunkt für den ursprünglichen NSGA die Evolutionsschleife. Srinivas und Deb verändern daran ausschließlich den Selektionsmechanismus. Dieser besteht einerseits aus dem in Abbildung 3.1 dargestellten Frontranking und andererseits aus einem Fitness-Sharing wie in Abschnitt 3.2.1 gezeigt. Das Fitness-Sharing-Schema unterstützt die Selektion bei der Qualitätsunterscheidung von Individuen innerhalb einer Front unvergleichbarer Individuen. Da zuallererst der Rang der Front, in der sich ein Individuum befindet, über den Fortpflanzungserfolg entscheidet, stellt das Fitness-Sharing eine Reduzierung der Fitness sicher, wenn sich viele Individuen in der gleichen Region der Front aufhalten. Die führt automatisch zu einer Reduzierung der Reproduktionswahrscheinlichkeit für die Individuen, die sich innerhalb der gleichen Nische eine Fitness teilen.

---

<sup>5</sup>Der Dominanzbereich einer Lösung ist derjenige Teil des Lösungsraumes, in dem alle Lösungen von der ersten Lösung dominiert werden.

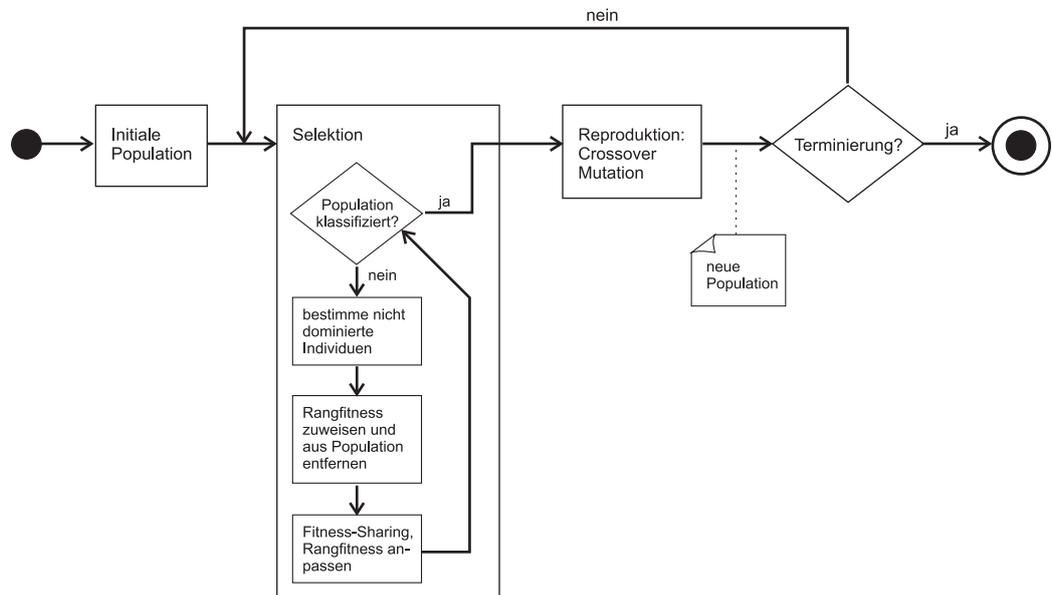


ABBILDUNG 3.2: Schematischer Ablauf von NSGA nach Srinivas & Deb (1994). Dabei unterscheidet sich die Evolutionsschleife nicht von der eines gewöhnlichen GAs. Jedoch kombiniert die Selektion das Frontranking mit einem Fitness-Sharing-Schema, um eine Bewertungsgrundlage für die Reproduktion zu schaffen.

Abbildung 3.2 stellt die Einbettung des angepassten Selektionsmechanismus in die Evolutionsschleife dar.

Für die erste Version von NSGA gilt damit, dass keine elitäre Selektion durchgeführt wird. Eine Population wird bewertet, und dann wird entsprechend der Reproduktionswahrscheinlichkeiten eine neue Population erstellt, die in der nächsten Generation allein die Grundlage für die Bewertung darstellt. Kritik an dieser Eigenschaft der ursprünglichen Umsetzung von NSGA, sowie die mit  $\mathcal{O}(MN^3)$  große Komplexität der dominanzbasierten Selektion<sup>6</sup> und die Notwendigkeit der Parametrisierung des Fitness-Sharing-Mechanismus begründeten schließlich den Vorschlag einer Anpassung des Algorithmus zu NSGA-2. In diesem verändern Deb et al. (2002) insbesondere folgende Aspekte:

1. Elitäre Selektion: Anders als in der ursprünglichen Variante von NSGA werden im neuen Algorithmus grundsätzlich Eltern und Nachkommen in den Selektionsprozess einbezogen. Dies führt zu einer Konservierung guter Lösungen in der Elternpopulation und größerer Konvergenzsicherheit.
2. Schnelleres Frontranking: Mithilfe eines angepassten Vorgehens beim Fron-

<sup>6</sup>Die Komplexität ergibt sich allerdings nur durch die Anzahl der nötigen Vergleiche in der Implementierung von NSGA. Durch einen relativ naiven Vergleichsmechanismus trägt die Bildung einer Front aufgrund wechselseitiger Vergleiche  $\mathcal{O}(MN^2)$  zur Laufzeit bei. Nimmt man im schlimmsten Falle an, dass jede Front nur ein Individuum enthält, so resultiert dies in der Abschätzung von  $\mathcal{O}(MN^3)$ .

tranking kann die Laufzeit auf  $\mathcal{O}(MN^2)$  reduziert werden. Dazu wird in  $\mathcal{O}(MN^2)$  Schritten jedem Individuum  $p$  die Anzahl  $n_p$  der Individuen zugeordnet, durch die es dominiert wird und die Menge  $S_p$  der Individuen, die es selbst dominiert. Aufgrund dieser Angaben ist es leicht, diejenigen Individuen zu identifizieren, die in der momentan nicht dominierten Front sind (alle Individuen mit  $n_p = 0$ ). Diese können temporär aus der Lösungsmenge entfernt werden. Anschließend kann für alle Individuen aus den Mengen  $S_i$  aller entfernten Individuen  $i$  der Dominanzzähler verringert werden. So ist das vorherige Vorgehen dann für die Bestimmung der nächsten Front ebenfalls anwendbar.

3. Anpassung der Nischenbildung und Diversitätserhaltung: Während die Fitness-Sharing-Methode aus NSGA noch abhängig von dem gewählten Nachbarschaftsradius war, führen Deb et al. (2002) mit der Crowding Distance-Metrik ein neues Nachbarschaftsmaß ein, das die Verteilung entlang einer Front nicht dominierter Lösungen abhängig von der direkten Nachbarschaft eines jeden Individuum bestimmt. Dabei wird für jedes Individuum  $k$  entlang jedes Kriteriums der normierte Abstand zwischen den direkten Nachbarn  $j$  und  $l$  bestimmt. Die Crowding Distance ergibt sich dann als Summe aller Teilergebnisse. Abbildung 3.3 stellt dieses Vorgehen für eine bikriterielle Lösungsmenge dar. Insgesamt kann dieses Vorgehen durch  $\mathcal{O}(N \log N)$  Schritte pro Kriterium umgesetzt werden, was für die Crowding Distance eine Komplexität von  $\mathcal{O}(MN \log N)$  ergibt. Die Crowding Distance wird in NSGA-2 als Ersatz für die geteilte Fitness im Falle der Qualitätsunterscheidung von Individuen gleichen Ranges genutzt.

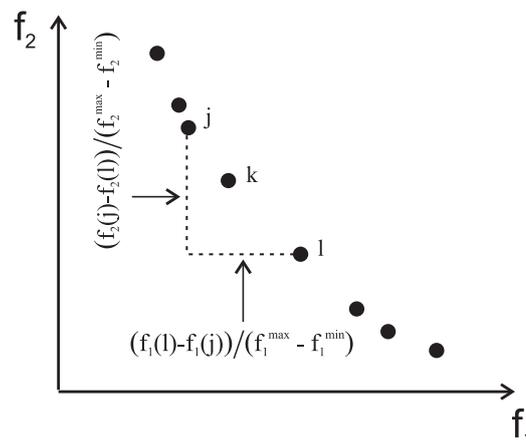


ABBILDUNG 3.3: Schematische Darstellung des Vorgehens zur Bestimmung der Crowding Distance einer Lösung  $k$  bezüglich ihrer direkten Nachbarn  $j$  und  $l$  im Lösungsraum. Die Metrik ergibt sich als Summe der normierten Abstände der Nachbarn bezüglich jedes Kriteriums.

Insgesamt stellt NSGA-2 inzwischen einen Quasi-Standard in dem Bereich der evolutionären mehrkriteriellen Optimierung dar. Fast jeder Algorithmus vergleicht

sich mit diesem Ansatz. Außerdem findet die Technik aufgrund der vielen bisher gesammelten experimentellen Erfahrungen auch im Kontext realer mehrkriterieller Optimierungsprobleme als relativ zuverlässige Strategie eine Anwendung<sup>7</sup>. Corne und Knowles haben jedoch auch Grenzen der dominanzbasierten Methoden – mit NSGA-2 als prominentem Vertreter – aufgezeigt (Knowles & Corne, 2007): mit zunehmender Anzahl der Kriterien verhält sich ein dominanzbasierter Ansatz nicht besser als eine Zufallssuche. Intuitiv bedeuten viele Kriterien für dominanzbasierte Ansätze, dass für eine kleine Population von Lösungen diese häufig unvergleichbar sind. Im Kontext von NSGA-2 bedeutet dies aber, dass sich alle Individuen in der Front mit Rang 1 befinden und die Selektion auf den die Diversität erhaltenden Mechanismus der Crowding Distance reduziert wird. Einen Ausweg aus dieser Situation bieten die indikatorbasierten Methoden des nächsten Abschnitts, deren Konvergenzvorteile man allerdings durch oftmals aufwändige Indikatorberechnungen erkauft.

### 3.2.2 Indikatorbasierte Methoden

Im Kontext der mehrkriteriellen Optimierung dienen Indikatoren als Maße zur Beurteilung einer Lösungsmenge bezüglich verschiedener Qualitätsmerkmale (eine genauere Darstellung erfolgt in Kapitel 7). Dabei wird einer Lösungsmenge ein skalarer Wert zugeordnet, der dann leicht mit den Qualitätswerten anderer Lösungsmengen verglichen und auf dieser Grundlage eine Beurteilung durchgeführt werden kann. Nutzt man diese Indikatoren als Ersatzziele für eine Optimierung, so kann man ein mehrkriterielles in ein einkriterielles Optimierungsproblem umformen und dies mit einem gewöhnlichen evolutionären Vorgehen behandeln.

Ein erstes allgemeines algorithmisches Rahmenwerk für diese Vorgehensweise und die Integration verschiedener Indikatoren schlugen Zitzler & Künzli (2004) mit dem sogenannten Indicator-Based Evolutionary Algorithm (IBEA) vor. Die Autoren betrachten für die Formulierung eines indikatorbasierten Selektionsmechanismus binäre Qualitätsmaße, die zwei Lösungsmengen vergleichen und eine Präferenz der einen über die andere ausdrücken. Insbesondere wenden sie die Indikatoren für die Fitnessbewertung jedes Individuums einer Population folgendermaßen an: Jedes Individuum der Population  $P$  wird mit allen übrigen Individuen der Population verglichen und aus allen Vergleichen zusammen ein Fitnesswert als

$$\nu(\vec{x}_i) = \sum_{\vec{x}_j \in P \setminus \{\vec{x}_i\}} -e^{-I(\{\vec{x}_j\}, \{\vec{x}_i\})/\kappa} \quad (3.4)$$

berechnet. Dabei ist  $I(\{\vec{x}_i\}, \{\vec{x}_j\}) < I(\{\vec{x}_j\}, \{\vec{x}_i\})$ , wenn  $\vec{x}_i$  die Lösung  $\vec{x}_j$  dominiert<sup>8</sup>. Der Wert  $\kappa$  ist ein indikator- und problemabhängiger Skalierungsfak-

<sup>7</sup>Am besten ist dies an der Tatsache abzulesen, dass NSGA-2 inzwischen als Optimierungsverfahren in der Global Optimization Toolbox von MatLAB zu finden ist, siehe <http://www.mathworks.de/products/optimization/>.

<sup>8</sup>Zitzler und Künzli betonen, dass dies nur mit dominanzerhaltenden Indikatoren funktioniert. Ein dominanzerhaltender binärer Indikator erfüllt:  $\vec{x}_1 \prec \vec{x}_2 \Rightarrow \nu(\vec{x}_1) > \nu(\vec{x}_2)$ .

tor (Zitzler & Künzli, 2004). In seiner algorithmischen Struktur unterscheidet sich der IBEA nicht von einer normalen Evolutionsschleife. Nach der Reproduktion (Reproduktionsselektion wird durch eine Turnierselektion realisiert) wird die nächste Generation von Individuen auf der Basis der berechneten Fitnesswerte der entstandenen Gesamtpopulation gebildet. Dabei werden sukzessive die Individuen mit geringster Fitness entfernt und anschließend die Fitnesswerte der verbliebenen Individuen in der Population durch einen Update-Schritt aktualisiert.

Zwei Algorithmen, die sich im Rahmenwerk von IBEA bewegen, aber doch eine sehr verschiedene Struktur und Umsetzung haben, sind der S-Metric Selection Evolutionary Multi-Objective Algorithm (SMS-EMOA) von Emmerich, Beume & Naujoks (2005) und der Hypervolume Estimation Algorithm (HypE) von Bader & Zitzler (2009, 2011). Beide sollen im Folgenden kurz diskutiert werden.

Der SMS-EMOA macht sich für die indikatorbasierte Selektion das durch eine Lösungsmenge und einen Referenzpunkt eingeschlossene Hypervolumen (Zitzler, Thiele, Laumanns, Fonseca & da Fonseca, 2003) im Zielfunktionsraum zunutze. Je größer dieses Volumen ist, das eine Lösungsmenge mit dem Referenzpunkt einschließt, desto besser ist die Approximation der Paretofront zu bewerten. Im Gegensatz zu IBEA greift SMS-EMOA jedoch nicht auf den Indikator als maßgebliche Komponente der Fitnessbewertung zurück, sondern integriert die Information in einen an NSGA angelehnten Mechanismus einer vorhergehenden Dominanzbewertung: Entsprechend eines Steady-State-Ansatzes wird durch einen Reproduktionsschritt ein einzelner Nachkomme erzeugt und damit die Populationsgröße um 1 erhöht. Als Selektionsschritt bleibt anschließend die Reduzierung der Population, damit die vorgegebene Populationsgröße wieder erreicht wird. Dazu werden zuerst die dominierten Lösungen der Population bestimmt. Existieren dominierte Lösungen, so wird einfach das Individuum mit der größten Dominanzzahl (siehe Individualranking in Abschnitt 3.2.1) entfernt. Gibt es in der untersuchten Population keine dominierten Individuen, so werden der Hypervolumenwert<sup>9</sup> der Population und damit zugleich der Beitrag jedes einzelnen Individuums dazu bestimmt. Das Individuum mit dem kleinsten Beitrag wird dann entfernt. Da das Hypervolumen gleichzeitig Konvergenz und Diversität einer Lösungsmenge quantifiziert, sichert dieser Schritt ein Konvergenzverhalten auch dann, wenn alle Individuen unvergleichbar sind, also im Sinne von NSGA-2 innerhalb einer Front liegen. Trotz dieses Vorteils und der prinzipiellen Fähigkeit, auch Probleme mit vielen Kriterien zu lösen, wird gerade dies durch den Rechenaufwand für die Bestimmung der Hypervolumenwerte erschwert. Zwar konnte Beume (2009) den Rechenaufwand bei einer Population  $P$  und  $m$  Kriterien von  $\mathcal{O}(|P|^{m-1})$  auf  $\mathcal{O}(|P| \log |P| + |P|^{m/2})$  reduzieren, für große  $m$  ist dies aber trotzdem oft der dominierende Faktor des Algorithmus.

Im Hypervolume Estimation Algorithm (HypE) adressieren Bader & Zitzler (2011) genau dieses Problem der Bestimmung des Hypervolumens für viele Kri-

---

<sup>9</sup>Im Kontext des Algorithmus und aus historischen Gründen wird das Hypervolumen oft auch als S-Metrik bezeichnet. Darin liegt die Namensgebung von SMS-EMOA begründet.

terien. Sie nutzen einen an IBEA angelehnten Algorithmus mit der Abfolge einer Reproduktionsselektion, Variation und anschließender Selektion zur Reduzierung der Gesamtpopulation. Dabei setzen sie insbesondere das Hypervolumen ein und gründen ihre Fitnessbewertung auf den Beitrag, den ein Individuum zum Gesamt-Hypervolumen der Lösung leistet („contributing hypervolume“). Die Bestimmung dieses Beitrags wird jedoch bei mehr als 3 Kriterien nicht exakt durchgeführt, sondern über einen Monte-Carlo-Ansatz approximiert. Ausgangspunkt ist die Partitionierung des Hypervolumens einer Lösungsmenge in Teile, die ausschließlich von einer Lösung und Teile die von mehreren Lösungen beigetragen werden. Für das gesamte Volumen wird dann der individuelle Beitrag einer jeden Lösung durch die Anzahl der zufällig in ihrer individuellen Partition platzierten Sample geschätzt, ebenso wie die gemeinsam beigetragenen Teile. Mittels dieser Informationen kann dann (ungefähr) bestimmt werden, welche Individuen viel und welche wenig zum Gesamtvolumen beitragen. Letztere werden während der Reduzierung der Gesamtpopulation entfernt. Durch die Schätzung des Hypervolumens kann die Laufzeit des Algorithmus auch für mehr als drei Kriterien stark beschleunigt werden und die Ergebnisse der Studien zeigen, dass der Ansatz bisherige exakte Methoden insbesondere in hochdimensionalen Zielräumen übertrifft.

### 3.2.3 Alternative Methoden

Natürlich finden sich in dem inzwischen großen Gebiet der evolutionären mehrkriteriellen Optimierung neben den zwei wichtigen Klassen der dominanz- und indikatorbasierten Algorithmen auch alternative Ansätze. Sie folgen auf den ersten Blick grundlegend verschiedenen und oftmals naturinspirierten Ideen, um Pareto-optimale Lösungen aufzufinden. Bereits der VEGA-Ansatz von Schaffer (1985) kann als alternative Methode zu den selektionsfokussierten Ansätzen der vorherigen Abschnitte betrachtet werden. So trennen auch Coello et al. (2007) die dominanz- und indikatorbasierten Methoden von sogenannten kriteriumbasierten Methoden. VEGA besitzt keine aufwändigen Selektionsmechanismen, sondern unterteilt die Population und selektiert darin jeweils einkriteriell entsprechend der unterschiedlichen Kriterien. Danach werden die „spezialisierten“ (Schaffer & Grefenstette, 1985) Ergebnisse der Teilbearbeitung durch Reproduktion zusammengeführt, so dass gute Kompromisslösungen entstehen können. Da die Reproduktion jedoch auf konvexen Operationen (Crossover) basiert und damit die genetischen Informationen der Individuen „kombiniert“ (Schaffer & Grefenstette, 1985) werden, entspricht dies dem Einsatz einer Gewichtung der Ziele, siehe Abschnitt 3.1.1, und ist somit nur auf konvexe Problemstellungen anwendbar. Aufgrund seiner einfachen algorithmischen Handhabbarkeit gilt dieser frühe Ansatz jedoch als Grundlage für viele ähnliche Konzepte (Kursawe, 1999). Auch das in dieser Arbeit untersuchte Räuber-Beute-Modell (Laumanns et al., 1998) orientiert sich an dieser Idee der algorithmischen Einfachheit.

Zusätzlich und exemplarisch für biologisch abweichend motivierte Ansätze soll hier noch detaillierter auf Arbeiten im Bereich der Schwarm-Ansätze (Particle

Swarm Optimization - PSO) für die mehrkriterielle Optimierung eingegangen werden: Die sogenannte Particle Swarm Optimization wurde von Kennedy & Eberhart (1995) als Optimierungskonzept – basierend auf dem sozialen Verhalten in einem Vogelschwarm – entwickelt. In Analogie zu der Beobachtung, dass Vogelschwärme bei der Futtersuche oftmals den Informationen von benachbarten Individuen folgen, formuliert man die Iterationsformel 1.6 aus Kapitel 1 mithilfe eines sogenannten Geschwindigkeitsvektors  $v_k$ , der sich aus den historischen Informationen über das betreffende Individuum selbst und den Informationen seiner Nachbarn zusammensetzt und damit eine Suchrichtung vorgibt.

$$\begin{aligned} \vec{x}_{k+1} &= \vec{x}_k + \vec{v}_{k+1}, \\ \text{mit } \vec{v}_{k+1} &= w \cdot \vec{v}_k + c_1 r_1 (\vec{x}_k^{best} - \vec{x}_k) + c_2 r_2 (\hat{x}^{global} - \vec{x}_k) \end{aligned} \quad (3.5)$$

Der Vektor  $\vec{x}_k$  beschreibt die Position des Individuums  $k$ . Die Geschwindigkeit wird gebildet, indem die durch einen Faktor  $w$  gewichtete vorherige Geschwindigkeit des Partikels und die historischen Informationen über die bisher beste Position  $\vec{x}_k^{best}$  des Partikels und die global beste Position  $\hat{x}^{global}$  innerhalb des gesamten Schwarms einbezogen werden. Die Konstanten  $c_1$  und  $c_2$  gewichten jeweils den Einfluss der entsprechenden Information,  $r_1$  und  $r_2$  sind Zufallszahlen im Intervall von  $[0, 1]$ . Im Unterschied zu evolutionären Algorithmen verfügt PSO damit nicht über explizite Selektionsmechanismen oder eine direkte Analogie zur Reproduktion. Vielmehr wird der kontinuierliche Flug eines Schwarms beschrieben, der der Richtung folgt, die ein Führungspartikel vorgibt (engl.: *leader*). Da diese Methodik jedoch insbesondere Probleme bei der Konvergenz aufweist und vorzeitige Stagnation in lokalen Optima nicht verhindert werden kann, wurde von Stacey, Jancic & Grundy (2003) die Einbeziehung von Mutation vorgeschlagen. Ihre Anwendung findet auf der neuen Partikelposition oder ausschließlich auf der Geschwindigkeit statt und ist inzwischen fester Bestandteil der PSO. Im Kontext der PSO wird die Mutation gewöhnlicherweise als *Turbulenz* bezeichnet.<sup>10</sup>

Der Transfer von PSO auf den mehrkriteriellen Fall stellt jedoch, wie bei den evolutionären Algorithmen, eine Herausforderung dar. Es muss nun eine Lösungsmenge bestimmt werden, womit sich automatisch die Frage nach der Darstellung des Führungspartikels und der Diversitätserhaltung stellt. Erstaunlicherweise finden sich auch hier Ansätze, die stark an die Vorgehensweisen evolutionärer Algorithmen erinnern: Parsopoulos & Vrahatis (2002) schlagen einen an VEGA angelehnten parallelen Mechanismus (VEPSO) vor, der mehrere Schwärme jeweils bezüglich eines einzelnen Kriteriums auswertet und dann die Informationen der global besten Individuen unter den Schwärmen austauscht. In einem anderen frühen Ansatz (MOPSO) von Coello Coello & Lechuga (2002) wird ein externes Archiv eingeführt, um nicht dominierte Lösungen zu speichern. Zugleich wird dieses Ar-

<sup>10</sup>Sicherlich wird dies in Anlehnung an atmosphärische Störungen und natürlich zur nachdrücklichen Betonung der Unterschiede zwischen Schwarmansätzen und evolutionären Methoden verwendet. Unabhängig von der abweichenden Namensgebung sind die Funktionsprinzipien von Mutation und Turbulenz jedoch gleich.

chiv als Grundlage für die Auswahl des Führungspartikels genutzt, indem der abgedeckte Suchraum aufgeteilt wird und die in den Teilen enthaltenen Individuen bewertet werden. Mittels einer Roulettrad-Selektion kann dann das Führungspartikel gewählt werden. Damit wird einerseits das Problem der Wahl des Führungspartikels, andererseits die Diversitätserhaltung adressiert. Die ausführliche Übersichtsarbeit von Reyes-Sierra & Coello Coello (2006) fasst mehrere Ansätze zusammen, die sich oftmals stark an den bekannten Mechanismen aus den evolutionären Algorithmen orientieren und daher an dieser Stelle nicht nochmal diskutiert werden sollen. Neuere Arbeiten im Bereich der mehrkriteriellen PSO beschäftigen sich verstärkt mit der Nutzung des Parallelisierungspotentials dieses Konzeptes (Scriven, Ireland, Lewis, Mostaghim & Branke, 2008) und nutzen so ein gewisses Alleinstellungsmerkmal dieser Ansätze gegenüber EAs.

Auch das in dieser Arbeit untersuchte Räuber-Beute-Modell deckt viele der bereits diskutierten Konzepte ab. Obwohl es zu den evolutionären Algorithmen gerechnet werden muss, liegt ihm eine erweiterte natürliche Motivation zugrunde, die auf die Interaktion von Spezies zurückgeht. Zudem wird in seiner frühen Variante bewusst auf komplexe Mechanismen zur Selektion und Diversitätserhaltung verzichtet und auf emergentes Verhalten der agierenden Einzelkomponenten gesetzt. Dies bringt schließlich auch ein großes Parallelisierungspotential mit.

Ausgehend von dem in diesem Teil erläuterten Hintergrund und den besprochenen deterministischen und randomisierten Verfahren zur Bearbeitung von mehrkriteriellen Optimierungsproblemen kann nun ein detaillierter Blick auf das Räuber-Beute-Modell geworfen werden. Untersucht wird dabei, inwieweit die von Laumanns et al. (1998) vorgeschlagenen Prinzipien anwendbar sind und wie sie gegebenenfalls verbessert werden können.

## Teil II

# Das Räuber-Beute-Modell für die mehrkriterielle Optimierung und die Untersuchung seiner Eigenschaften



## Kapitel 4

# Die Idee des Räuber-Beute-Modells

Betrachtet man die algorithmische Entwicklung von heuristischen Verfahren zur Optimierung, in denen der Zufall als explorative Komponente eine zentrale Rolle spielt, so fällt auf, dass oftmals naturnahe Modelle, Systeme oder Prozesse als Motivation oder Namensgeber fungieren. So existieren neben den in Kapitel 2 vorgestellten Verfahren, die sich allesamt auf das Prinzip der Evolution berufen, verschiedene andere Bezeichnungen für spezielle Ansätze, die etwa die Analogie zum Ameisenstaat (Dorigo & Stützle, 2004), zum Verhalten des Vogelschwarms (Kennedy & Eberhart, 1995), der Funktion des menschlichen Immunsystems (Dasgupta & Attouh-Okine, 1997; Castro & Timmis, 2002) oder zu allgemeinen biologischen Prinzipien wie etwa der Koevolution (Hillis, 1990; Popovici, Bucci, Wiegand & de Jong, 2011) betonen. Dieser Trend kann zwei Motivationen haben:

1. Einerseits liefert die Analogie zur Natur eine sprichwörtlich „natürliche“ Intuition für die Funktionsweise des Algorithmus und befreit zudem bis zu einem gewissen Grad vom Zwang zur Erklärung der genauen Dynamik, die der Methode zu Grunde liegt<sup>1</sup>. Denn bereits aus der Beobachtung der Ergebnisse natürlicher Prozesse ist ja intuitiv klar, dass diese grundsätzlich „funktionieren“, wenn man sie zwar abstrakt aber analog zur Natur umsetzt. Dabei ist auch zu beachten, dass viele der vorgebrachten Analogien natürlich auf einer sehr makroskopischen Betrachtung der Evolution oder anderer natürlicher

---

<sup>1</sup>Besonders herausragende Beispiele der kreativen aber ausnahmsweise nicht naturnahen Namensgebung für Algorithmen sind etwa in den Arbeiten von Ursem (1999) oder Thomsen, Rickers & Krink (2000) zu finden. Schlägt ersterer noch eine Analogie zu Nationen, Regierungen und Politik vor, um spezielle Populationsstrukturen und algorithmische Dynamik zu beschreiben, verlegt sich die zweite Gruppe von Autoren auf eine Analogie mit Religionen. In ihrem Religion-based EA, der seine Ursprünge angeblich im sogenannten Jihad-EA hat, werden Paarungsbeschränkungen und Migration zwischen Teilpopulationen durch Religionszugehörigkeit bzw. Missionierung/Konvertierung dargestellt. Außer einem kuriosen Beitrag zur Benennung von Algorithmen beschränkt sich die Untersuchung in den Arbeiten jedoch auf die erstaunte Beobachtung und Dokumentation von bereits bekannten evolutionären Effekten.

Prozesse basiert. In den über 150 Jahren seit der Veröffentlichung von Darwins Theorie hat sich diese jedoch selbst stark verändert. Das Verständnis oder eine detaillierte Erklärung der zugrunde liegenden Mechanismen – sowohl in der biologischen wie auch in der technischen Betrachtung – ist jedoch oft schwierig. So ist es manchmal einfacher Emergenz – also die fast magische Herausbildung von gewünschten Eigenschaften eines Systems als Konsequenz des Zusammenwirkens von Teilkomponenten – oder einfache evolutionäre Prozesse anzunehmen und sogleich auf eine detaillierte Betrachtung der inneren Prozesse des Systems zu verzichten.

2. Andererseits kann die modellhafte Betrachtung natürlicher Vorgänge und die Übertragung des biologischen Prozessverständnisses durchaus zu wichtigen Einsichten bezüglich der einem Algorithmus innewohnenden Prinzipien beitragen. Doch auch der umgekehrte Weg ist denkbar: Das Verständnis eines aus der Biologie motivierten, technischen Prozesses kann zur Vertiefung des Verständnisses des biologischen Prozesses selbst beitragen.

Es ist aber sicher festzustellen, dass der Fokus bei der Benennung der meisten Optimierungsverfahren auf den ersten Fall abzielt. Nur wenige Autoren (Schwefel & Kursawe, 1998; Rudolph & Schwefel, 2008) fordern in diesem Bereich gleichfalls eine Beschäftigung mit dem Gegenstand des zweiten Punktes. Die einfache Erklärung des algorithmischen Prinzips spielt vordergründig und historisch auch für das in dieser Arbeit besprochene und untersuchte Räuber-Beute-Modell eine Rolle. Die aus der Biologie entlehnte Analogie der Wechselbeziehung zwischen Räubern und Beute wird dort als eine Ausprägung des Prinzips der Koevolution betrachtet, bei der sich mehrere Spezies gegenseitig durch individuelle Fortentwicklung einem Evolutionsdruck aussetzen (Campbell & Reece, 2006). Konkret für die Räuber-Beute-Beziehung bedeutet dies: Verändert sich etwa die Beute-Spezies durch evolutionäre Weiterentwicklung, so führt dies zu einer Bedrohung der Existenz der Räuber (durch Verhungern), wenn diese sich nicht auf die neuen Eigenschaften ihrer Beute anpassen können. Dies löst dann einen entsprechenden Selektionsdruck aus und führt zu einer evolutionären Weiterentwicklung der Räuber. Umgekehrt löst eine evolutionäre Veränderung der Räuber eine Bedrohung der Existenz der Beutepopulation aus, so dass auch hier der Druck für eine evolutionäre Anpassung erhöht wird.

Teile dieser Wechselwirkung wurden von Laumanns et al. (1998) als algorithmisches Prinzip auf die mehrkriterielle Optimierung übertragen. Sie verkürzen das ursprüngliche biologische Prinzip jedoch insofern, dass die Entwicklung der Räuberindividuen und damit auch die Koevolution (im biologischen Sinne) aufgegeben wird. So wird die biologische Räuber-Beute-Analogie – wie oben dargestellt – ein plakatives Sinnbild des algorithmischen Prinzips. Anders als zuvor angesprochen, beobachten die Autoren in ihren ersten Arbeiten (Laumanns et al., 1998; Laumanns, 1999) viele Aspekte des Algorithmus sehr ausführlich und liefern erste Einblicke in das Systemverhalten. Sich dran anschließende Arbeiten anderer

Autoren (Deb, 2001; Li, 2003; Schmitt, Mehnen & Michelitsch, 2005) bauen jedoch erstaunlicherweise nicht auf diesen Einsichten auf, sondern nutzen vielmehr oberflächliche Beobachtungen und die Analogie zum Räuber-Beute-Modell als Ausgangspunkt für fast willkürliche Modifikation durch Mechanismen aus der ein- oder mehrkriteriellen Optimierung.

Als Einstieg und zur Motivation später durchgeführter formaler Betrachtungen wird in diesem Kapitel auf die entsprechenden Vorarbeiten eingegangen und einerseits die algorithmische Idee des Räuber-Beute-Modells für die mehrkriterielle Optimierung vorgestellt. Andererseits werden einige grundlegende Strukturelemente, Aktoren und Konzepte im Algorithmus betrachtet und begründet. Auf Grundlage durchgeführter Beobachtungen am Basissystem werden dann die von anderen Autoren durchgeführten Modifikationen beschrieben und bewertet.

## 4.1 Schwefels Räuber-Beute-Modell

Geht man bei der Betrachtung eines mehrkriteriellen Optimierungsproblems von der in Definition 1.6 angenommenen Aufteilung in  $m$  Zielfunktionen aus, so ist jede einzelne Zielfunktion ein Kriterium, das ein Individuum im Sinne der evolutionären Optimierung gut erfüllen muss, um zu überleben. Unabhängig davon wie diese Zielfunktion genau aussieht, stellt sie eine Einflussgröße im Kontext der Evolutionsmechanismen eines EA dar und ihre Auswertung kann für jedes Individuum als Bedrohung verstanden werden. Da es in dem MOP mehrere solcher Bedrohungen gibt, kann man leicht die Analogie eines Räuber-Beute-Verhältnisses zwischen Zielfunktion und Lösung formulieren. In dieser Vorstellung repräsentieren die Zielfunktionen Prädationseinflüsse, also Räuber, während die Population von Lösungen bedrohte Beuteindividuen darstellt. Damit ist im Prinzip die grundlegende Erwartung von Laumanns et al. (1998) formuliert: Durch den Einfluss verschiedener Räuber (also Zielfunktionen) ist ein Beuteindividuum gezwungen, sich möglichst gut an alle Räuber anzupassen, um zu überleben. Damit ist laut den Autoren zu erwarten, dass sich Kompromisse zwischen den jeweiligen Zielfunktionen im Laufe des Evolutionsprozesses automatisch herausbilden.

Um der natürlichen Analogie nahe zu kommen und einen genetischen Austausch unter Beuteindividuen nur in enger Nachbarschaft zu erlauben, wird von Schwefel und seinen Kollegen auf eine zelluläre Populationsstruktur (siehe Abschnitt 2.2.4) zurückgegriffen, genauer auf einen zweidimensionalen Torus. Dieser ermöglicht einerseits die unendliche Bewegung auf der Graphenstruktur und stellt zugleich sicher, dass jeder Knoten des Graphen für  $t \rightarrow \infty$  unendlich oft und gleich häufig erreicht werden kann (Laumanns et al., 1998).

Geht man von dieser Struktur und den darauf unbeweglich platzierten Beuteindividuen aus, so besteht die algorithmische Umsetzung aus der Bewegung und Aktion der Räuber, siehe auch Abbildung 4.1. Jeder Räuber bewegt sich mittels eines Zufallslaufes (engl. *random walk*) entlang den Kanten der Graphenstruktur. Abhängig von der Anzahl der durchgeführten zufälligen Schritte erreicht der

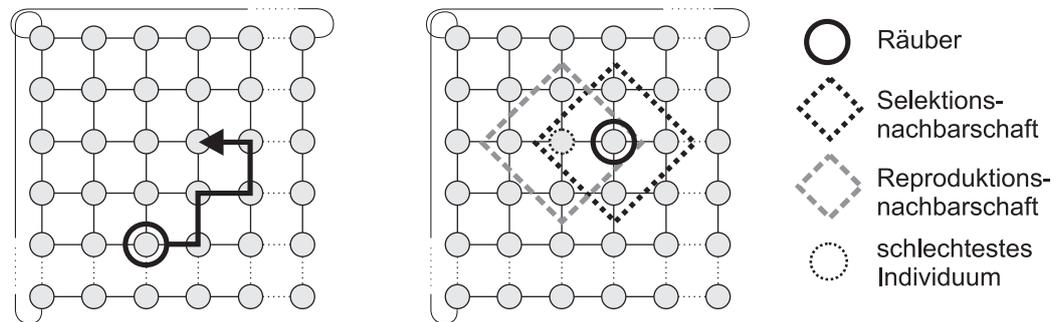


ABBILDUNG 4.1: Schematische Darstellung der wichtigen Elemente und Schritte des Räuber-Beute Modells. Nachdem der Räuber eine zufällige Bewegung auf der toroidalen Graphenstruktur durchgeführt hat, wird innerhalb der Selektionsnachbarschaft das schlechteste Individuum bzgl. des Räuberkriteriums bestimmt. Ein potentieller Nachkomme wird aus der Reproduktionsnachbarschaft um das schlechteste Individuum herum erzeugt.

Räuber schließlich eine Position auf dem Graphen, um die herum eine Selektionsnachbarschaft aufgespannt wird.<sup>2</sup> Alle darin enthaltenen Individuen werden in den Bewertungsprozess bezüglich des vom Räuber repräsentierten einzelnen Kriteriums einbezogen. Das am schlechtesten bewertete Individuum fällt schließlich dem Räuber zum Opfer und muss aus benachbarten Individuen ersetzt werden. Dazu wird um den potentiell freien Platz auf der Graphenstruktur eine Reproduktionsnachbarschaft aufgespannt, innerhalb derer aus der vorhandenen genetischen Information der übrigen Beuteindividuen durch Mutation und/oder Rekombination ein neues Individuum erzeugt wird.

Bereits bei dieser informalen Konstruktion ergeben sich einige Parameter, über die das Verhalten des Modells verändert werden kann. Neben der Schrittweite der Räuberindividuen beim Zufallslauf, sind auch die Größen der Selektions- und Reproduktionsnachbarschaften einstellbar. Mit zunehmender Größe der Räuberschnittweite verliert die Räuberaktion ihre Lokalität: Wählt man die Schrittweite sehr klein ( $=1$ ), so bewegt sich der Räuber kaum und die anschließend aufgespannten Nachbarschaften können sogar Elemente enthalten, die bereits in einem vorherigen Schritt betrachtet wurden. Bei einer sehr großen Schrittweite springt der Räuber praktisch zufällig von einer Position auf dem Torus zu einer beliebigen anderen Position. Einen ähnlichen Einfluss hat die Wahl der Größe der Selektions- und Reproduktionsnachbarschaften. Wird der Radius der Nachbarschaft klein gewählt, so werden nur wenige Individuen in den Selektionsprozess oder in den Reproduktionsvorgang einbezogen und die Effekte sind auf eine Subpopulation begrenzt. Bei einem großen Nachbarschaftsradius wird ein Großteil oder sogar die gesamte Population einbezogen. Dies entspricht schließlich der Selektion auf panmiktischen Populationsstrukturen (siehe Abschnitt 2.2.4) und steuert

<sup>2</sup>In der Arbeit gehen wir hierbei wie Laumanns et al. (1998) von einer von-Neumann-Nachbarschaft aus.

der zuvor festgelegten Strukturierung der Population des Räuber-Beute Modells entgegen. Hier ist dann die Position des Räubers bedeutungslos.

Natürlich sind auch die Mutations- und Rekombinationsarten, die Mutationsschrittweite sowie die Anzahl der Räuber eines jeden Kriteriums Faktoren, die auf die Lösungsbildung wirken. In ihrer ersten Arbeit zum Räuber-Beute Modell haben Laumanns et al. (1998) und wenig später auch Laumanns in seiner Diplomarbeit (Laumanns, 1999) einige dieser Faktoren untersucht und experimentelle Beobachtungen beschrieben. Die Wichtigsten sind:

1. Die Größe der Selektions- und Reproduktionsnachbarschaft, die Schrittweite der Räuber und die Anzahl der Räuber haben einen ähnlichen Einfluss auf die Lösungsbildung. Werden die Faktoren klein gewählt, so tendieren die Lösungen zu den Extremregionen der Paretomenge, also zu den besten Lösungen der Einzelkriterien. Dabei geht die Diversität der Lösungsmenge zurück; Kompromisslösungen in der Mitte der Paretomenge werden nicht gefunden oder gehen über die Zeit verloren. Werden die Faktoren groß gewählt, so werden insbesondere Lösungen in der Mitte der Paretomenge gefunden. Extremlösungen gehen weitgehend verloren. Auch Untersuchungen anderer Autoren (Deb, 2001; Ai, 2005; Grimme, 2005; Grimme, Lepping & Papasprou, 2007) stellen ein solches Verhalten des Modells fest. Beispielhaft sind die zwei Fälle dieses Lösungsbildungsverhaltens für unterschiedliche Nachbarschaftsgrößen in Abbildung 4.2 für die in Beispiel 1.11 eingeführte mehrkriterielle Kugelfunktion  $\mathcal{F}_{\odot}^2$  analog zu den Resultaten von Grimme et al. (2007) dargestellt. Die Darstellung für verschiedene Schrittweiten der Räuber wäre analog.
2. Die verteilte Population bildet teilweise feste Unterpopulationen ähnlicher Individuen, die auch durch das Eindringen eines Räubers mit gegensätzlichem Kriterium nicht stark verletzt werden. Hier vermuten Laumanns et al. (1998) dass dieses Verhalten an der fest gewählten Mutationsschrittweite liegt und fordern einen Adaptationsmechanismus, um diesem Effekt entgegenzuwirken. Diese Beobachtung und die Ergründung ihres Ursprungs ist, wie sich in dieser Arbeit noch zeigen wird, einer der wichtigsten Aspekte zum Verständnis der Dynamik des Räuber-Beute Modells. Erstaunlicherweise wurde er von Laumanns nur in einem Nebensatz erwähnt und später nicht weiter untersucht. Zu einem späteren Zeitpunkt wird der Effekt in dieser Arbeit ausführlich besprochen und die Wirksamkeit der von Laumanns und Kollegen geforderten Anpassung der Mutationsschrittweite widerlegt.
3. Die in Verbindung mit Mutation durchgeführte Rekombination hat oftmals einen negativen Einfluss auf das Konvergenzverhalten des Modells. Auch spätere Untersuchungen von Ai (2005) und Grimme (2005) haben diese Ergebnisse bestätigt.

Interessant ist, dass diese Beobachtungen direkt als Motivation für verschiedene Erweiterungen des Räuber-Beute Modells genutzt wurden. Ohne die Effekte

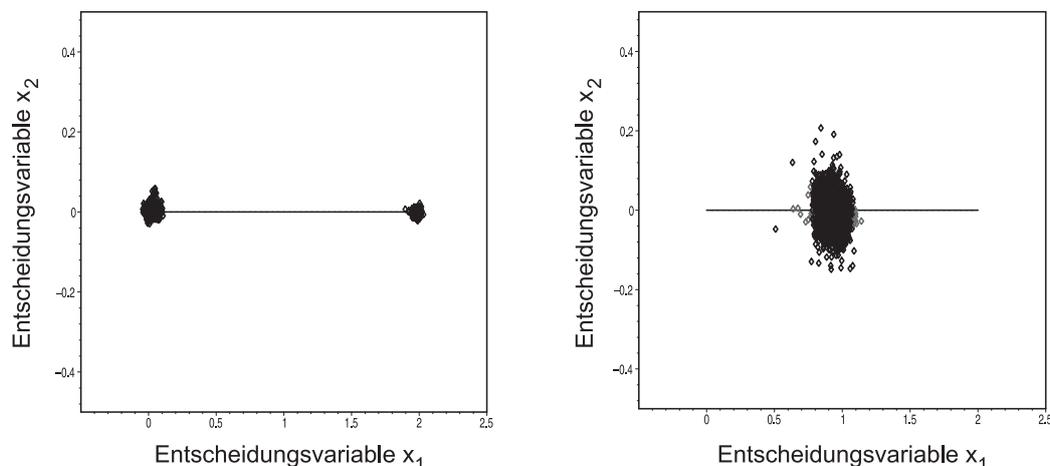


ABBILDUNG 4.2: Beispielhafte Darstellung der Paretomenge des mehrkriteriellen Kugelmodells  $\mathcal{F}_{\odot}^2$ , ermittelt mit dem grundlegenden Räuber-Beute Modell mit zwei Räubern, geringer Mutationsschrittweite, einschränktem Zufallslauf des Räubers. Für die linke Darstellung wurde eine Selektions- und Rekombinationsnachbarschaft von 1, auf der rechten Seite wurden die Nachbarschaften entsprechend der Gesamtpopulation gewählt (Grimme et al., 2007). Die eingezeichnete Linie repräsentiert die analytisch bestimmte Paretomenge.

im Einzelnen erklären zu können, haben verschiedene Autoren das Modell adaptiert und durch zusätzliche Mechanismen versucht, die beobachteten aber nicht verstandenen Effekte zu beheben oder zu kompensieren. Dabei wurde oft auf die ersten Erklärungsansätze von Laumanns et al. (1998) oder die etwas detailliertere Untersuchung aus Laumanns Diplomarbeit (Laumanns, 1999) zurückgegriffen. Schon deshalb liegt der Fokus der durch andere Autoren vorgeschlagenen Erweiterungen des Modells insbesondere auf der Erzeugung von Zwischenlösungen und der Integration von Adaptationsmechanismen für die Mutationsschrittweite.

## 4.2 Varianten und verwandte Ansätze

Das Räuber-Beute Modell wurde zunächst von Deb (2001) in seiner Monographie über Ansätze zur evolutionären mehrkriteriellen Optimierung aufgegriffen und leicht verändert. Ausgehend von der Beobachtung, dass das originale Modell nicht in der Lage ist, eine ausreichende Diversität der Lösungsmenge herzustellen, schlägt Deb vor die einkriterielle Selektion so zu verändern, dass jeder Räuber nach einer individuellen Gewichtung und Aggregation aller Kriterien selektiert. Entsprechend der Gewichtungsmethode aus Abschnitt 3.1.1 soll ein Räuber so einen speziellen Abschnitt der Paretofront ermitteln. Die Verwendung mehrerer Räuber mit unterschiedlichen Gewichtungen führt dann zur gleichzeitigen Approximation verschiedener Teilgebiete der Paretofront. Analog zur Diskussion der Gewichtungsmethode ist auch bei diesem Vorgehen festzustellen, dass das Modell durch diese Erweiterung nicht die Fähigkeit gewinnt, nicht-konvexe Teile der

Paretomenge zu erreichen. Zudem verlangt der Ansatz einen Mechanismus, um die Gewichtungen der Kriterien innerhalb jedes Räubers zu wählen oder anzupassen. Hier wurden keine Vorschläge gemacht. Insgesamt zeigte sich, dass die Ergebnisse von Deb sich neben einer etwas beschleunigten Konvergenz nicht signifikant von jenen des originalen Modells unterschieden. Bei sehr einfachen und natürlich konvexen Testproblemen konnte jedoch eine bessere Diversität erreicht werden. Viel erstaunlicher ist jedoch, dass bereits in einer frühen Arbeit der eigentliche Gedanke des von Schwefel vorgeschlagenen Modells aufgegeben wurde, ohne die innewohnende Dynamik oder beobachtete Schwächen verstanden zu haben. Deb reduziert den Ansatz auf ein Multiagentensystem in dem lediglich verschiedene Instanzen der Gewichtungsmethode parallel laufen.

Eine noch umfassendere Veränderung des Modells wurde 2003 von Li (2003) vorgeschlagen. Er erlaubt nicht nur die Bewegung von Räubern sondern ebenfalls von Beuteindividuen. Beute bewegt sich nun auch zufällig über den Torus. Nach einer Bewegung versucht die Beute eine Fortpflanzungsphase mit benachbarten Individuen einzuleiten. Dies funktioniert jedoch nur, wenn sich in definierter Nähe andere Beuteindividuen befinden.<sup>3</sup> Die Räuber bilden wie bei Laumanns et al. und Deb die Bedrohung für die Beuteindividuen und laufen ebenfalls zufällig über den Torus. Jedoch führt Li eine Iterationshäufigkeit für jeden Räuber ein, die bestimmt, wie oft ein Räuber pro Beute-Schritt jagen darf. Diese Häufigkeit richtet sich nach der Differenz zwischen der aktuellen und gewünschten Anzahl von Beuteindividuen in Relation zu den vorhandenen Räubern (Li, 2003), um die Ausrottung aller Beuteindividuen durch die Räuber zu verhindern. Abgesehen davon, dass Li durch seine Erweiterungen keine signifikanten Verbesserungen der Approximationsergebnisse erreichen kann, verfügt diese Erweiterung über verschiedene neue Parameter und damit über neue Einflussgrößen auf die Dynamik des bis dahin weitgehend unverstandenen Räuber-Beute-Ansatzes. So kann die Analyse von Li und die Erweiterung sicherlich als ein erster Beitrag zur experimentellen Erforschung des Modells angesehen werden. Eine Erklärung für das Verhalten und eine gut begründete Modifikation liefert sie nicht.

Schmitt et al. (2005) wenden sich dem Räuber-Beute Modell zu und knüpfen an Laumanns' Versuche zur Anpassung der Mutationsschrittweite an (Laumanns, 1999). Hatte Laumanns die Mutation noch entsprechend einer monoton fallenden Funktion im Laufe des Evolutionsprozesses angepasst, so streben die Autoren die Integration des Selbstadaptationsmechanismus der Evolutionsstrategien nach Schwefel (1995) an. Dieses Anpassungsverfahren erlernt evolutiv die nötigen Strategieparameter für die Mutation. Es setzt aber für seine gute Anwendbarkeit einen Überschuss an Nachkommen während der Reproduktion – also einen hohen Selektionsdruck – voraus. Da das Räuber-Beute Modell nach Laumanns jedoch als Steady-State-Verfahren bezeichnet werden kann, müssen Schmitt und Kollegen erhebliche Veränderungen vornehmen. Die zelluläre Strukturierung der Individuen

---

<sup>3</sup>Li geht in seinem Ansatz von einer sogenannten Moore-Nachbarschaft aus, die die acht umliegenden Positionen mit einbezieht.

wird in sogenannte Deme unterteilt, in denen sich Räuber weiterhin zufällig bewegen. Jedoch ist ein Räuber nun einem speziellen Gebiet zugeordnet. Gleichzeitig wird zufällig bestimmt, in welchem relativen Verhältnis die Räuber innerhalb eines Gebietes jagen. In jedem Dem jagen die Räuber solange, bis eine Mindestanzahl an Beuteindividuen übrig ist. Aus diesen Individuen werden dann die Nachkommen erzeugt. Dabei werden neben der normalen Mutation und Rekombination ebenfalls die von den Beuteindividuen getragenen Strategieparameter verändert und im Sinne des Selbstadaptationsmechanismus aus den Evolutionsstrategien weitergegeben. Die anschließend durchgeführten Untersuchungen zeigen, dass der Mechanismus in der Lage ist, konvexe Probleme mit besserer Diversität als das originale Räuber-Beute Modell zu lösen. Jedoch scheitert auch dieser Ansatz an der Approximation nicht-konvexer Probleme. Diese Erkenntnis entspricht etwa den zuvor beschriebenen Ergebnissen von Deb (2001). Der Einfluss der eingeführten Selbstadaptation wird von den Autoren leider nicht untersucht oder diskutiert. Dies ist relativ überraschend, da für dieses Vorgehen verschiedene, die Systemdynamik stark beeinflussende Änderungen vorgenommen werden. Eine ausreichende Begründung für all diese Modifikationen fehlt völlig.

Noch schwerwiegender sind jene Modifikationen, die Deb & Udaya (2005) am Räuber-Beute Modell durchführen. Zuerst ersetzen die Autoren den einkriteriellen Selektionsmechanismus durch einen elitären und dominanzbasierten Bewertungsmechanismus. Darin wird ein Beuteindividuum nur ersetzt, wenn der Nachkomme das entsprechende Individuum zumindest schwach dominiert. Außerdem wird ein diversitätserhaltender Mechanismus eingeführt. Bei der Selektion eines Nachkommen wird zusätzlich zur Dominanzeigenschaft überprüft, ob die erzeugte Lösung bezüglich des Zielfunktionsraumes in der Umgebung eines bereits existierenden Beuteindividuum liegt. Ist dies der Fall, so wird die erzeugte Lösung nicht akzeptiert. Die Größe der Umgebung der Individuen ist extern einzustellen. Obwohl dieser Ansatz grundsätzlich zu guten Ergebnissen bei konvexen und nicht-konvexen Problemen führt, ist die Motivation für deren Durchführung am Räuber-Beute Modell nicht ersichtlich. Der neue Selektionsmechanismus entspricht weitgehend dem Vorgehen, das bei NSGA-2 (Deb et al., 2002) und ähnlichen dominanzbasierten Verfahren angewendet wird. Mit seiner Einführung verliert das Räuber-Beute Modell einige seiner wichtigen Vorteile. Neben der Einfachheit des Selektionsmechanismus geht auch die Entkopplung der einzelnen Prozesse verloren. Bei jeder Selektion muss die gesamte Population betrachtet werden, um die die Diversität erhaltende Bewertung anzuwenden. In diesem Kontext ist nicht einsehbar, welchen direkten Beitrag die Arbeit zum Verständnis des Räuber-Beute-Modells liefert und warum dieser Ansatz als Erweiterung des Modells betrachtet werden sollte. Er ist vielmehr eine grundlegende Veränderung des algorithmischen Prinzips.

Schließlich soll im Zusammenhang der Erweiterungsanstrengungen des Räuber-Beute Modells noch die Arbeit von Drezewski & Siwik (2007) genannt werden. In ihrem Ansatz wird das von Laumanns vorgeschlagene Modell als Motivation für die Übertragung zuvor entwickelter ko-evolutionärer Techniken für die einkriterielle Optimierung verwendet (Drezewski, 2003): Unter Beibehaltung einiger

Aspekte, wie der verteilten Population und der grundsätzlichen Unterscheidung zwischen Räuber- und Beuteindividuen, entwickeln die Autoren einen komplexen Interaktionsmechanismus zwischen diesen sogenannten Agenten. Dazu wird den Beuteindividuen wie bei Li (2003) die Bewegung auf der räumlichen Struktur der Population erlaubt. Zusätzlich werden sogenannte Ressourcen eingeführt, um die die Beute untereinander, sowie Räuber und Beute konkurrieren. Diese Ressourcen werden ausgetauscht, wenn ein Räuber eine Beute „frisst“. Zum Austausch kommt es aber auch, wenn zwei Beuteindividuen aufeinander treffen. Es werden Ressourcen von einer Beute an eine andere Beute abgegeben, wenn die zweite die erste dominiert, oder die Individuen zu nah bei einander im Zielfunktionsraum liegen. Dieses Vorgehen ist ähnlich zu dem Selektionsprozess von Deb & Udaya (2005). Weiterhin bestimmen die Ressourcen (über eine einstellbare Schwelle) neben den Geschlechtern zweier Beuteindividuen, ob sich diese fortpflanzen dürfen. Für einige Probleme können die Autoren zeigen, dass diese vorgeschlagenen Modifikationen bessere Ergebnisse als der einfache Ansatz von Laumanns und sogar als die Anwendung von NPGA (siehe Abschnitt 3.2.1) erreichen. Bei einigen anderen geprüften Problemen sind jedoch die beiden Referenzalgorithmen besser. Zum Verständnis der Dynamik tragen auch diese Modifikationen nicht bei. Vielmehr wird durch sie weitere Dynamik hinzugefügt, die es annähernd unmöglich macht die einzelnen Aspekte des Modells zu analysieren.

### 4.3 Ein formales und modifiziertes Modell

Wie zuvor gezeigt, fehlt in der Literatur eine konkrete und kritische Untersuchung einzelner Systemaspekte des Räuber-Beute-Ansatzes. Allen Arbeiten fehlt eine ausreichende Einsicht in die beobachteten Effekte, um darauf aufbauend Änderungen begründet vorzuschlagen oder das Modell zu verwerfen. Entweder werden für die Modifikationen hergebrachte Methoden angewendet, die die Dynamik des Räuber-Beute Modells zumindest teilweise überlagern, oder es wird viel neue Dynamik hinzugefügt, die sich formal nur sehr schwer beherrschen lässt. Erst die Arbeiten von Grimme, Lepping, Papaspyrou und Schmitt (Grimme, 2005; Grimme & Schmitt, 2006; Grimme & Lepping, 2007; Grimme et al., 2007), die mit ihren zentralen Aussagen auch Bestandteil dieser Arbeit sind, legen die Grundlage für ein Verständnis der Prozesse innerhalb des Räuber-Beute-Modells und damit für die Modifikation des Algorithmus und seine spätere Übertragung auf kombinatorische Probleme. Jene zentralen Ergebnisse werden im nächsten Kapitel dargestellt und verallgemeinert. Für die Handhabung der Dynamik des Räuber-Beute Modells ist es aber notwendig, das zuvor von Laumanns et al. (1998) definierte Modell leicht anzupassen. Dies betrifft insbesondere die *Äquivalenz von Selektions- und Rekombinationsnachbarschaften*, sowie die *Kopplung von Räubern und Operatoren*. Es sei jedoch betont, dass diese geringen Änderungen – wir diskutieren sie im Folgenden detaillierter – keine Auswirkung auf die Gesamtdynamik des Modells haben. Sie vereinfachen jedoch die Analyse signifikant.

Formal liegt allen hier betrachteten Instanzen des Räuber-Beute Modells eine toroidale Graphenstruktur  $G = (V, E)$  mit der Knotenindexmenge  $V = \{1, \dots, v^2\}$  und Kantenmenge  $E$  zugrunde. Für die Knoten gilt, dass sie auf einem Einheitsgitter angeordnet sind. Der Wert  $v$  ist dabei die Seitenlänge des Gitters. Für alle  $i, j \in V$  und die Kantenmenge gilt, dass die Kante  $(i, j) \in E$  ist, wenn für Knoten  $i$  an der Gitterposition  $(a, b)$ ,  $0 \leq a \leq v - 1$ ,  $0 \leq b \leq v - 1$  gilt, dass Knoten  $j$  an den Gitterpositionen  $((a \pm 1) \bmod v, b)$  oder  $(a, (b \pm 1) \bmod v)$  ist (Zuckerman, 1992).

Weiterhin kann eine Menge  $\mathfrak{B}$  von Beuteindividuen definiert werden. Folgt man dem Ansatz Laumanns', so wird auf jedem Knoten des Torus  $G$  ein Beuteindividuum platziert (Laumanns et al., 1998), es gilt also  $\text{card}(V) = \text{card}(\mathfrak{B})$ . Diese repräsentieren die Lösungskandidaten für ein betrachtetes MOP und werden über die Zeit durch das Räuber-Beute-Modell verändert. Das MOP  $\mathcal{F} = (f_1, \dots, f_m)$  verfolgt gleichzeitig  $m$  Ziele, die jeweils Teil einer Räuberspezies sind. Den zweiten Teil eines Räubers stellt der einem Räuber zugeordnete Variationsoperator  $O : \mathfrak{B} \rightarrow \mathfrak{B}$  dar. So ergibt sich formal die Räubermenge  $\mathfrak{R} := \mathcal{F} \times O = \{(f_c, op) \mid f_c \in \mathcal{F}, op \in O\}$ . Die Räuber verfügen zudem über eine Bewegungsfunktion  $walk : V \rightarrow (V \rightarrow [0, 1])$ , die jeder Bewegung von einem Knoten zu einem anderen Knoten eine Wahrscheinlichkeit zuordnet. Im betrachteten Modell nehmen wir immer einen gleichverteilten einschrittigen Zufallslauf an, so dass sich die konkrete Instanziierung

$$walk(v_t)(v_{t+1}) = \begin{cases} \frac{1}{d(v_t)} & , \text{ wenn } (v_t, v_{t+1}) \in E \\ 0 & , \text{ sonst.} \end{cases}$$

ergibt. Dabei ist  $v_t$  der Ausgangs- und  $v_{t+1}$  der Zielknoten der Bewegung, während  $d(v_t)$  den Grad des Knotens bezeichnet. So wird jedem direkten Nachbarn des Ausgangsknoten die gleiche Wahrscheinlichkeit zugeordnet während des Zufallslaufes erreicht zu werden. Ebenso werden vom Räuber die Selektions- und die Reproduktionsnachbarschaft als dieselbe Menge  $\mathfrak{N}(v, r) \subset V$  mit  $v \in V$  und  $r \in \mathbb{N}$  aufgespannt, die all jene Elemente enthält, die von Knoten  $v$  aus mit  $r$  Schritten erreichbar sind:  $\mathfrak{N}(v, r) = \bigcup_{(v, v') \in E} \mathfrak{N}(v', r - 1)$ . Die Anzahl der Knoten in einer Nachbarschaft mit Radius  $r$  ist als

$$|\mathfrak{N}(v, r)| = (r + 1)^2 + r^2 \tag{4.1}$$

gegeben<sup>4</sup>. Schließlich bleibt für die Vervollständigung des formalen Modells nur noch die Beschreibung des Räuberhaltens, der einzig aktiven Komponente des Modells. Algorithmus 1 fasst dies zusammen.

Solange kein Terminierungskriterium eintritt (hier wird oftmals eine maximale Anzahl von Räuberschritten verwendet), bewegt sich der Räuber zuerst entsprechend seines Zufallslaufes von Position  $\pi_i$  auf eine neue Position  $\pi_{i+1}$ . Dort werden

<sup>4</sup>Oftmals wird diese Nachbarschaft als von-Neumann-Nachbarschaft bezeichnet. Die Anzahl der in der Nachbarschaft enthaltenen Knoten ergibt sich über  $1 + \sum_{k=1}^r 4k = 1 + 4 \sum_{k=1}^r k = 1 + 4 \frac{r(r+1)}{2} = (r + 1)^2 + r^2$ .

---

**Algorithmus 1** Allgemeines Räuberverhalten.

---

**Eingabe:**  $\mathfrak{R} \ni \mathfrak{r} = (f_c, op), v_0, r, \pi_0$

```

1: while NOT (Terminierungskriterium erfüllt) do
2:   Zufallslauf  $\pi_i \rightarrow \pi_{i+1}$ 
3:    $SN = RN = \mathfrak{N}(\pi_{i+1}, r)$ 
4:   bewerte Beute in  $SN$  bzgl.  $f_c$ , markiere schlechteste  $b_{worst} \in \mathfrak{B}_{SN}$  für spätere
     Ersetzung
5:   Nachkomme  $b_{off} := Op(RN \setminus b_{worst})$ 
6:   bewerte den Nachkommen  $b_{off}$  bzgl.  $f_c$ 
7:   if  $b_{off} \leq b_{worst}$  then
8:     ersetze  $b_{worst}$  durch  $b_{off}$ 
9:   end if
10: end while

```

---

die Selektionsnachbarschaft (SN) und die Reproduktionsnachbarschaft (RN) aufgespannt. Dies ist eine der Modifikationen im Vergleich zu Laumanns' Modell: in unserem Ansatz unterscheiden sich die Nachbarschaften nicht. In der Selektionsnachbarschaft wird jede Beute bzgl. Kriterium  $f_c$  ausgewertet und die schlechteste markiert. Nach der Erzeugung eines Nachkommens aus den übrigen verfügbaren Individuen und mithilfe des an den Räuber gekoppelten Operators, wird bestimmt, ob dieser die schlechteste Beute ersetzt und die Ersetzung gegebenenfalls durchgeführt. Die eingeführte Äquivalenz der Selektions- und Reproduktionsnachbarschaft sorgt für eine bessere Analysierbarkeit des Ersetzungsvorgangs. Im Gegensatz zu Laumanns' Modell ist bei diesem Ansatz die Qualität aller Individuen bekannt, die an dem Ersetzungsprozess innerhalb der Nachbarschaft beteiligt sind. Die Bindung eines Variationsoperators an einen Räuber ermöglicht zudem das lokale Einbringen von verschiedenen Einflüssen. So kann durch jeden Räuber ein anderer, eventuell sogar mit dem Räuberziel verbundener Operator in die lokale Evolution eingebracht werden. Bei genauer Betrachtung umfasst dieses Vorgehen zumindest alle Teilaspekte der von Laumanns und Kollegen vorgeschlagenen ursprünglichen Version der Variation. Zwar war dort die Variation allgemein an die Beute gebunden, jedoch wurde in der Literatur niemals ein Szenario evaluiert in dem jede Beute über unterschiedliche Reproduktionsmechanismen verfügt. Alle bisherigen Umsetzungen führen nach dem Entfernen einer Beute eine direkte Reproduktion aus. Dieses Verhalten kann weiterhin auch mit der hier vorgeschlagenen Anpassung abgebildet werden.

Durch beide vorgenommenen Änderungen lässt sich der Reproduktionsprozess in der späteren Betrachtung vollständig und aus der Sicht des Räubers modellieren. Betrachten wir den Selektionsoperator und die Variation jeweils als einen Baustein des Räubers, so kann entsprechend der Darstellung von Grimme und Lepping Grimme & Lepping (2007) jeder Räuber als beweglicher Agent in der Populationsumgebung mit verschiedenen Kombinationen aus Selektions- und Variationsoperatoren eingesetzt werden, siehe Abbildung 4.3.

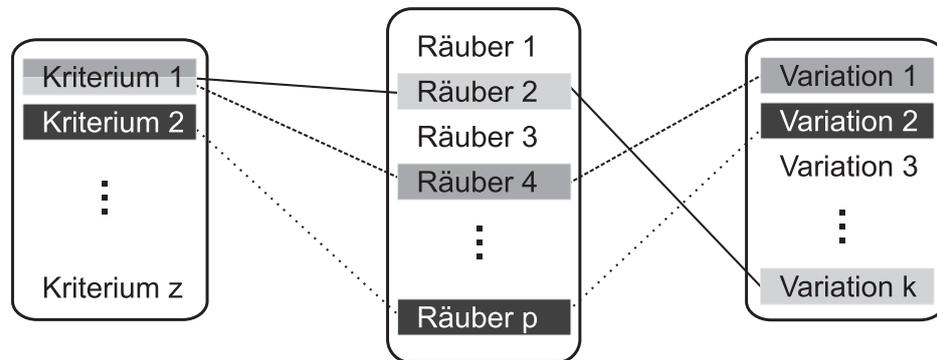


ABBILDUNG 4.3: Konzeptionelle Darstellung der Kombination von Selektion und Variation als verschiedene Räuberspezies, um spezielle lokale Effekte in die Populationsentwicklung zu bringen, nach Grimme & Lepping (2007).

Jede dadurch entstehende Kombination von Kriterium und Variation kann wieder als Räuberspezies betrachtet werden. Wie in Abbildung 4.4 dargestellt, lassen sich auf diese Weise verschiedene lokale Variationseinflüsse – gekoppelt an ein spezielles Selektionskriterium – in den Evolutionsprozess und die implizite Interaktion der Räuber einbringen. Dieser modulare Aufbau des Räubers aus unterschiedlichen Bausteinen ist zudem Grundlage für die im nächsten Kapitel durchgeführte Analyse der Einflüsse von Selektion und Mutation. Weiterhin ist diese Eigenschaft ein wichtiger Ausgangspunkt für die Integration von Expertenwissen wie sie im letzten Teil dieser Arbeit behandelt wird.

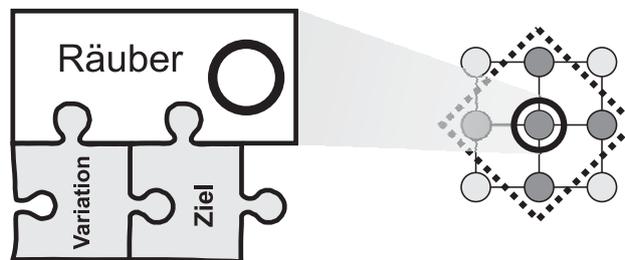


ABBILDUNG 4.4: Schematische Darstellung des Aufbaus eines Räuberindividuums, bestehend aus dem einzelnen verfolgten Ziel und dem gekoppelten Variationsoperator. Dieser wird auf die lokale Nachbarschaft zur Reproduktion eines Individuums nach der Selektion angewendet.

## Kapitel 5

# Analyse von Eigenschaften des Räuber-Beute-Modells

Das Räuber-Beute-Modell verfügt in seiner definierten Struktur über zwei wichtige Eigenschaften, die seine Untersuchung und insbesondere seinen praktischen Einsatz motivieren. Einerseits können die räumlich verteilte Population und seine unabhängig agierenden Räuber zur Parallelisierung des Algorithmus genutzt werden. Andererseits kann der modulare Aufbau der Räuber und des restlichen Modells genutzt werden, um an verschiedenen Stellen problemspezifische Anpassungen einzubringen, die dann in ihrer Kombination auf der verteilten Population ein gemeinsames Resultat erzeugen. Diese Erwartung war implizit auch mit dem originalen Modell Laumanns' verbunden, deren Entstehungsprozess muss jedoch genau beleuchtet werden. In diesem Kapitel soll, ausgehend von dem zuvor definierten formalen Modell, zuerst der Aspekt der Parallelisierung betrachtet werden. Für die weitere Diskussion der Erweiterungen des Räuber-Beute-Modells von zentraler Bedeutung ist die Untersuchung der Auswirkung der Räuber auf die Beutepopulation. Sie bildet schließlich den Ausgangspunkt für eine systematische Modifikation des Modells.

### 5.1 Exkurs zur Modularität und Parallelisierbarkeit des Räuber-Beute-Modells

Im Kontext nicht-linearer Optimierung wird oftmals die gute Parallelisierbarkeit evolutionärer Algorithmen als wichtiges Argument für den Einsatz dieser populationsbasierten Heuristiken angeführt (Alba & Tomassini, 2002; Cantú-Paz, 2005). Dabei ist diese Eigenschaft der Parallelisierbarkeit grundsätzlich in der Verwendung der Population begründet und erlaubt direkt die nebenläufige Auswertung aller Individuen (*engl.: embarrassingly parallel execution*). Jedoch ist die Parallelisierbarkeit durch den zentralen Auswertungsmechanismus eines normalen evolutionären Algorithmus begrenzt. In einer panmiktischen Population folgt der Auswertung grundsätzlich ein Selektionsschritt, in dem aufgrund der erzeugten Ergebnisse

über die Zusammensetzung der nächsten Generation von Individuen entschieden wird. An dieser Stelle müssen alle Ergebnisse vorliegen, so dass hier eine Synchronisationsstelle oder Barriere existiert, an der auf die Fertigstellung jeder nebenläufigen Auswertung gewartet werden muss. Bei sehr unterschiedlichen Laufzeiten der Auswertungen kann dies zu einer starken Einschränkung des Parallelisierungsvorteils führen. Trotzdem ist dieses Vorgehen auch im Falle der mehrkriteriellen evolutionären Optimierung einer der am häufigsten verwendeten Ansätze zur Parallelisierung, da er fast keine Anpassungen am vorhandenen Algorithmus erfordert. Powell & Hollingsworth (2007) parallelisieren NSGA-2 direkt durch den Einsatz von Webservices für die Durchführung von Funktionsauswertungen. Die Dominanzsortierung und Crowding Distance-Berechnung bilden natürliche Synchronisationsstellen. Die Autoren ermitteln einen annähernd linearen Geschwindigkeitsgewinn (Speedup), gehen dabei jedoch von vollständig gleichartigen Laufzeiten für einzelne Auswertungen aus. Mit der durchgeführten einfachen Anpassung gilt dies aber tatsächlich nur bei dem angenommenen Szenario mit annähernd gleichartigem Aufwand pro Funktionsauswertung. Bei der Optimierung industrieller Prozesse – in denen eine Funktionsauswertung oftmals aus dem Aufruf komplexer Simulationsumgebungen besteht – findet sich dieser Fall gewöhnlicherweise nicht. Hier kann das Warten auf die Fertigstellung einer sehr aufwändigen Auswertung zu einer ungünstigen Ausnutzung der verfügbaren parallelen Ressourcen führen. Jedoch ist dieser Fall in der Regel nicht schlechter als die sequentielle Ausführung.

Soll die Nebenläufigkeit von Prozessen innerhalb eines Algorithmus erhöht und damit die Parallelisierbarkeit verstärkt werden, so kann dies meist nur durch eine Restrukturierung des ursprünglichen evolutionären Algorithmus erfolgen. Für die monolithischen Verfahren der mehrkriteriellen evolutionären Optimierung bedeutet dies die Aufgabe der globalen Betrachtung von Lösungen sowie eine Einschränkung der Anwendung von Dominanzrelation und diversitätserhaltenden Mechanismen. Auch die Rückkopplung von Archiven in den Evolutionsprozess wird damit eingeschränkt. Gewöhnlicherweise wird hier zur Umsetzung auf eine stärkere Strukturierung der Population gesetzt. Dies ist schließlich gleichbedeutend mit der Unterteilung der Population in verschiedene Gebiete, in denen unabhängig von einander evolutionäre Suchprozesse ablaufen. Solche Gebiete werden oft als Inseln bezeichnet und etwa bei Xiong & Li (2003) für SPEA2, von Okuda, Hiroyasu, Miki & Watanabe (2002) für den gleichzeitigen Einsatz von ein- und mehrkriteriellen Verfahren oder von Parsopoulos, Tasoulis & Vrahatis (2004) für den gleichzeitigen Betrieb mehrerer Schwarminstanzen eingesetzt. Gewöhnlicherweise löst man die Verbindung zwischen den Inseln nicht vollständig auf, sondern erlaubt – wie in der zuvor genannten Literatur – eine spärliche Migration von Lösungen, um damit den globalen Zusammenhang der Population zu erhalten und das „Einwandern“ innovativer Lösungen in Inselpopulationen zu erlauben. Ein Gesamtergebnis entsteht dann aus der Gesamtheit der ermittelten Teilergebnisse.

Einige Klassifikationen von parallelen ein- und mehrkriteriellen Algorithmen (Alba & Tomassini, 2002; Coello et al., 2007) unterscheiden noch den Inselansatz und dessen spezielle Ausprägung als Diffusionsmodell. Sie betrachten das Diffusi-

onsmodell mit seiner zellularen Populationstruktur (siehe Kapitel 2.2.4) als eigene Klasse der Parallelisierung. Etliche Ansätze (Siirola, Hauan & Westerberg, 2004; Rowe, Vinsen & Marvin, 1996), darunter auch das Räuber-Beute-Modell, verfolgen diesen Ansatz. Stärker als im allgemeinen Inselmodell wird der evolutionäre Suchprozess nebenläufig und nur noch lokal auf einer Nachbarschaft weniger Individuen ausgeführt, wobei die Individuen selbständig als einzelne Agenten agieren oder sich als unabhängig arbeitende Agenten über die Population bewegen und ihre Operationen ausführen. Informationen werden so nur noch indirekt durch die Überschneidung von Nachbarschaften propagiert. Eine konkrete Interaktion von Prozessen (wie bei der Migration im Inselmodell) findet nicht statt. Auf diese Weise trägt die Strukturierung der Population zu einer ausgeprägten Entkopplung und zur Realisierung nebenläufiger Prozesse bei.

Zur weiteren Einordnung des Räuber-Beute-Modells soll an dieser Stelle auf eine Klassifikation von Talbi, Mostaghim, Okabe, Ishibuchi, Rudolph & Coello Coello (2008) eingegangen werden, die die Parallelisierung mehrkriterieller Verfahren aus einer eher konzeptionellen und problembezogenen Perspektive und nicht von der Populationsstruktur ausgehend betrachtet. Dazu führen die Autoren drei Ebenen der Betrachtung ein:

1. Die oberste Ebene der parallelen Kooperation umfasst sowohl die Populationsstruktur wie auch die konzeptionelle Ausprägung des parallelen Algorithmus. Insbesondere wird hier die Interaktion zwischen einzelnen algorithmischen Modulen betrachtet. Dabei bedeutet der Term der Kooperation nicht nur direkte Kooperation im engen Sinne von aktivem Informationsaustausch oder algorithmischer Beeinflussung. Kooperation meint insbesondere auch die indirekte Zusammenarbeit und Beeinflussung auf einer gemeinsamen Population oder Datenstruktur.
2. Die mittlere Ebene betrifft die *problemunabhängige* Parallelisierung im Sinne eines Master-Slave-Verfahrens, wie es zuvor für die nebenläufige Durchführung von Funktionsauswertungen beschrieben wurde. Unabhängig vom betrachteten Problem können voneinander entkoppelte Aufgaben ausgeführt werden und so eine Beschleunigung der Gesamtbearbeitung erreicht werden.
3. Die dritte Ebene fokussiert schließlich anwendungsspezifisch auf die Unterteilung von einzelnen Evaluationen, um diese zu beschleunigen. Nach Talbi et al. (2008) können auf dieser Ebene *problemabhängig* vielerlei Ansätze verfolgt werden: mehrere Kriterien können etwa mit ganz unterschiedlichen Suchstrategien angegangen, einzelne Auswertungen unterteilt und parallelisiert oder auch mehrere Evaluationen unter Unsicherheit (z.B. Rauschen auf den Ergebnissen der Funktionsauswertungen) für das Erreichen robuster Ergebnisse parallel ausgeführt und später zu einem Ergebnis aggregiert werden.

Abbildung 5.1 stellt schematisch die Einordnung des Räuber-Beute-Modells in die von Talbi und Kollegen vorgeschlagene Klassifizierung dar. Bei dieser Betrachtung

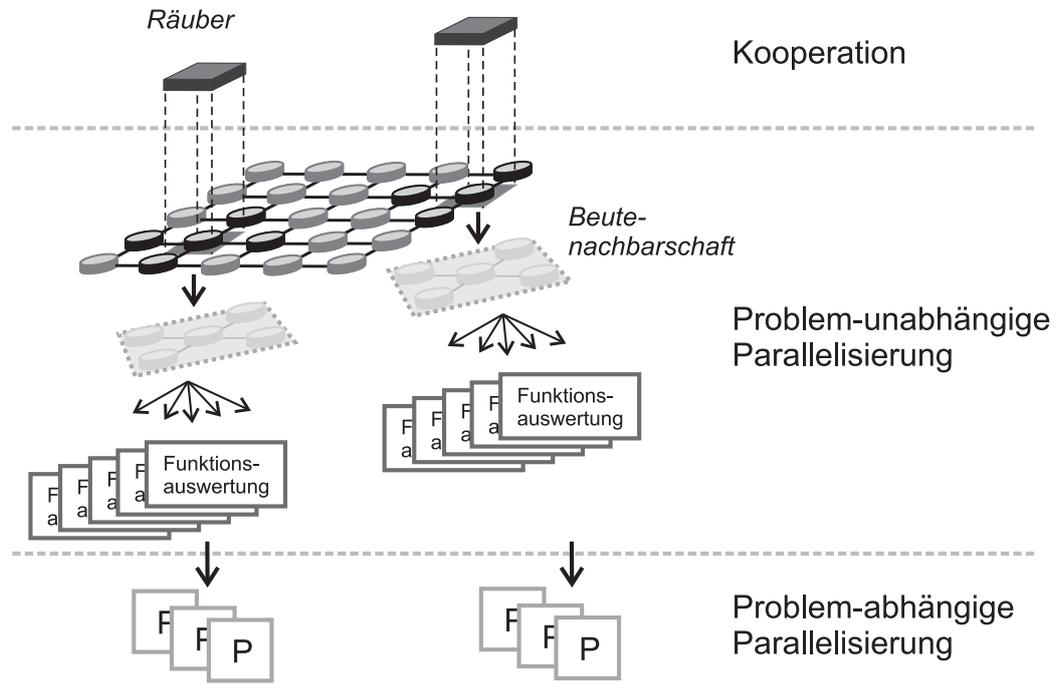


ABBILDUNG 5.1: Einordnung des betrachteten Räuber-Beute-Modells in das von Talbi et al. (2008) vorgeschlagene Schema zur Betrachtung von Parallelisierungsaspekten eines mehrkriteriellen Algorithmus, nach Grimme et al. (2011).

tung ist festzustellen, dass das Räuber-Beute-Modell auf allen drei Ebenen ein erhebliches Parallelisierungspotential ausschöpft. Auf der obersten Ebene der parallelen Kooperation sind die Räuberbewegungen auf der verteilten Population als integraler Teil des Algorithmus anzuordnen. Durch ihre Aktivität beeinflussen die Räuber die Beutepopulation bezüglich ihres einzelnen Kriteriums und nur durch die implizite Kooperation mehrerer Räuber mit unterschiedlichen Kriterien wird schließlich eine Lösungsbildung für das mehrkriterielle Problem erwartet. Hierbei arbeiten die Räuber vollständig entkoppelt, und wissen gegenseitig von ihrer Existenz nicht. Global gesehen arbeiten sie aber in indirekter Kooperation auf einer gemeinsamen Population und beeinflussen sich durch ihren lokalen Einfluss auf eine kleine Nachbarschaft von Beuteindividuen gegenseitig bei der Lösungsfindung. Aus der Sicht der Parallelisierung bringt dieses emergente Verhalten einen wichtigen strukturellen Vorteil mit sich. Es wird dadurch, im Vergleich zu vielen etablierten und monolithischen Verfahren, eine implizite Parallelisierung der dominanzbasierten Selektion erreicht, die sich pro Akteur ausschließlich auf die Betrachtung kleiner Umgebungen bezüglich eines Kriteriums bezieht. Durch diese lokale Beschränkung verschwindet die Barriere für nebenläufige Funktionsauswertungen und eine anschließenden globalen Bewertung fast vollständig. Das nicht vollständige Verschwinden der Barriere ergibt sich durch eventuelle Überschneidungen von Beuteregionen zweier Räuber und der Größe der Nachbarschaft. Stre-

ben zwei Räuber die Ersetzung der gleichen Beute an, so ist es notwendig Synchronisierungsmechanismen für den gemeinsamen Zugriff auf die entsprechenden Objekte einzusetzen.<sup>1</sup> Außerdem entsteht nach wie vor eine Barriere durch die parallel Auswertung der lokalen Nachbarschaft. Die Bearbeitung anderer Nachbarschaften wird davon jedoch nicht beeinflusst.

Auf der zweiten Ebene bleibt die Möglichkeit der problemunabhängigen Parallelisierung in zweierlei Ausprägung. Auch durch die lokale Selektion auf Ebene der Räuberkooperation entsteht durch die Betrachtung einer kleinen Nachbarschaft von Individuen um die Räuberposition herum eine Subpopulation, die für einen Vergleich ausgewertet werden muss. Diese Auswertungen können stark parallelisiert werden. Dabei entsteht natürlich eine Auswertungsbarriere, die jedoch aufgrund der unabhängig agierenden Räuber und der relativ kleinen Nachbarschaft meist vernachlässigt werden kann. Als zweiter Aspekt der problemunabhängigen Parallelisierung lässt sich an dieser Stelle die feste Platzierung der Beute betrachten. Jedes Beuteindividuum kann – nebenläufig zu den Aktionen der Räuber – selbst eine lokale Optimierung der individuellen Lösungsqualität durchführen, etwa durch deterministische lokale Suchverfahren. Später wird auf diesen Aspekt nochmals eingegangen, wenn Gradientenverfahren als lokale Suchstrategien in das Räuber-Beute-Modell und auf Ebene der Beuteindividuen integriert werden, siehe Kapitel 7.

Auf unterster Ebene kann schließlich die Parallelisierung des Problems selbst durch angepasste Auswertungsmethoden betrachtet werden. Hat das betrachtete Optimierungsproblem keine mathematische Formulierung, basiert die Evaluation oftmals auf Simulationen, die für die spezielle Problemklasse angepasst sind. Simulationen realer Systeme sind häufig sehr komplex und zeitaufwändig, so dass vielfach eine Parallelisierung und damit Beschleunigung des Simulationsvorgangs angestrebt wird. Spezielle Anwendungsprogramme wie etwa CFD-ACE+<sup>2</sup> und ANSYS CFX<sup>3</sup> sowie Simulationsumgebungen wie MATLAB<sup>4</sup> unterstützen automatische aber problemabhängige Dekomposition für die parallele Auswertung. Natürlich ist die Ausnutzung der problemabhängigen Parallelisierung kein expliziter Vorteil des Räuber-Beute-Modells. Die Betrachtung zeigt jedoch, dass der Ansatz auch auf der dritten Ebene durch seine Konzepte keine Beschränkung einführt. So können moderne Mehrkernarchitekturen für sehr kommunikationsintensive, da stark gekoppelte Parallelisierung genutzt werden, während die höheren Parallelisierungsebenen mit wenig Kommunikation auskommen und Interprozessorkommunikation oder sogar auf Ebene ganzer Rechencluster nebenläufig effizient arbeiten können.

Die vorhergehende Betrachtung des Räuber-Beute-Modells unter dem Aspekt der Parallelisierung zeigt insbesondere, dass die algorithmische Struktur durch die

---

<sup>1</sup>Im einfachsten Fall sperrt der zuerst zugreifende Räuber den Zugriff auf das Objekt. Für diesen Zeitraum ist die Aktivität des anderen Räubers blockiert.

<sup>2</sup><http://www.esi-group.com/>

<sup>3</sup><http://www.ansys.com/>

<sup>4</sup><http://www.mathworks.com/>

Entkopplung auf der Ebene der Kooperation der Räuber zu einem Auflösen von Parallelisierungsschranken führt. Dabei werden die beiden, von annähernd allen evolutionären Verfahren umgesetzten Ebenen der problemunabhängigen und problemabhängigen Parallelisierung, nicht beeinflusst. Da gerade die Kooperations-ebene eine wichtige Rolle in diesem Algorithmus spielt wird im weiteren Verlauf der Arbeit die Dynamik der Kooperation der entkoppelt agierenden Räuber und ihr Beitrag zur Lösungsbildung im mehrkriteriellen Falle untersucht.

## 5.2 Aspekte der Interaktion von Räuber- und Beute-population

Bereits die Beobachtungen von Laumanns et al. (1998) und die von anderen Autoren durchgeführten Modifikationen zeigen, dass die eigentlich unabhängig voneinander agierenden Räuber gemeinsam auf die Beutepopulation wirken und so gemeinsam zur Lösungsbildung beitragen. Die Diskussion in Kapitel 4 hat aber auch gezeigt, dass dieses Zusammenspiel der Räuber noch tiefergehend ergründet werden muss. Aufbauend auf dem in Abschnitt 4.3 definierten formalen und leicht modifizierten Modell soll nun eine detaillierte Analyse der Systemdynamik und der indirekten Kooperation der Räuber vorgenommen werden. Diese Analyse wird kleinschrittig und ausgehend von den Haupteffektoren, den modularen Räubern, durchgeführt. Der modulare Aufbau eines jeden Räubers – bestehend aus einkriterieller Selektion und Variationsoperator – hilft bei der feingranularen Analyse verschiedener Einflüsse. So kann das Zusammenwirken von Räubern unterschiedlicher Kriterien für Mutation und Rekombination getrennt betrachtet werden. Entsprechend gliedert sich auch dieser Abschnitt in einige Vorbereitungen zur Analyse, die anschließende Betrachtung des Mutationseinflusses, die Untersuchung des Effekts der Rekombination und schließlich in eine gemeinsame Betrachtung der Variations-einflüsse.

### 5.2.1 Einige Vorbereitungen

Die genaue Betrachtung der Populationsdynamik erfordert einen Einblick in die Populationsstruktur und in den Vorgang der Lösungsbildung. Es reicht dabei nicht aus, lediglich die Ergebnisse einer Simulation wie üblich über Bewertungsmaße zu betrachten. Vielmehr muss beobachtet und später beschrieben werden, wie sich Lösungen im Räuber-Beute-Modell über die Zeit und unter den gegebenen Einflussfaktoren herausbilden. Hier ist insbesondere interessant, wie sich die nicht mobile Beutepopulation unter dem Einfluss von Räuber-induzierter Mutation bzw. Rekombination verhält. Ein Weg dieses Verhalten zu beobachten, ist die direkte Visualisierung der Lösungsqualität jedes Individuums in der Population. Um dies sinnvoll umzusetzen, müssen jedoch einige Beschränkungen eingeführt werden:

1. Wir betrachten speziell das Problem  $\mathcal{F}_{\odot}^2$  aus Beispiel 1.11, da dies einerseits eine bekannte Lösungsmenge besitzt und andererseits für eine Analyse hin-

reichend einfach ist. Wie in Abbildung 5.2 dargestellt besteht das Problem aus zwei Kriterien, die jeweils durch eine (verschobene) Kugelfunktion repräsentiert werden. Die optimalen Lösungen der einzelnen Kriterien liegen bei  $(0, 0)$  und  $(2, 0)$ . Die Lösungsmenge im Suchraum beschreibt alle Punkte auf der Linie zwischen den Extrempunkten:  $x_1 \in [0, 2]$  und  $x_2 = 0$ . Der Konflikt der beiden Kriterien liegt also insbesondere in der  $x_1$ -Komponente. Beide Kriterien verfolgen bzgl. der  $x_2$ -Komponente das gleiche Ziel. Daher gehen wir davon aus, dass die  $x_2$ -Komponente in dem Vorgang der Kompromissbildung eine neutrale Rolle spielt. Somit schränken wir unsere Analyse auf den Konflikt der Kriterien ein und betrachten ausschließlich die Ausdehnung der Lösungsmenge in  $x_1$ -Richtung. Dies bedeutet, dass in der späteren Visualisierung eine Aussage über die Nähe zu den optimalen Positionen im Entscheidungsraum nur für die Komponente  $x_1$  getroffen werden kann. Eine Aussage über die  $x_2$ -Komponente der Lösungen ist nicht möglich.

2. Zudem vernachlässigen wir für die Analyse der Dynamik jegliche Methoden zur Archivierung von bereits erreichten Lösungen. Obwohl es für den praktischen Einsatz des Algorithmus eine natürliche Maßnahme wäre, bereits gefundene gute Lösungen aufzubewahren, wäre ein solcher Mechanismus bei der Analyse der Systemdynamik hinderlich, da er den tatsächlichen Zustand der Lösungsmenge zumindest teilweise verdecken würde. Diese Vernachlässigung kann jedoch ohne Einfluss auf den Algorithmus angenommen werden, da das Räuber-Beute-Modell keine Rückkopplung eines Archivs in den Evolutionsprozess vorsieht.

Mit diesen Annahmen ist es möglich die Entwicklung der Lösungsbildung in der verteilten Population zu visualisieren. Dazu definieren wir eine Farbskala für die Lösungen der Paretomenge des Problems  $\mathcal{F}_{\odot}^2$ : Sei  $\mathcal{C} = (c_1, \dots, c_l, c_{l+1}, c_{l+2})$  eine diskrete Farbskala mit  $l + 2$  Farbwerten. Wir unterteilen nun die die Lösungsmenge repräsentierende Gerade mit  $0 \leq x_1 \leq 2$  und  $x_2 = 0$  in  $l$  Intervalle, und belegen jedes Intervall mit einem der  $l$  Farbwerte. Zwei Farben  $c_{l+1}$  und  $c_{l+2}$  werden dann allen rechts und links von der Gerade liegenden Lösungen zugewiesen. Sie markieren so Lösungen außerhalb des betrachteten Kompromissbereichs, siehe Abbildung 5.2. Der Übersichtlichkeit halber wählen wir diese Farben gleich den Farben der jeweiligen Extremlösungen.<sup>5</sup>

Weiterhin definiere  $\mathcal{Q} = \{q_{ij} \mid i, j \in \{1, \dots, k\}\}$  ein in  $k^2$  gleiche Teilquadrate unterteiltes Quadrat. Wir wählen  $k$  entsprechend der Seitenlänge des Torus, der die Populationsstruktur bestimmt. Jedes Teilquadrat  $q_{ij} \in \mathcal{Q}$  beschreibt dann eine Beuteposition auf dem Torus. Färbt man schließlich die Teilquadrate aus  $\mathcal{Q}$  mit Farbwerten der Farbskala  $\mathcal{C}$ , so ergibt sich eine Einordnung der Beute bezüglich

---

<sup>5</sup>Diese Vereinfachung verändert unsere qualitativen Beobachtungsergebnisse nur geringfügig. Ergebnisse die bzgl. der  $x_1$ -Komponente jenseits der Extremlösungen liegen, sind nach wie vor relativ gut für ein Kriterium und sehr schlecht für das andere. Eine genauere Betrachtung der Menge führen wir später während der Analyse der Ausdehnung der konvexen Hülle durch. Diese Methode dient nur der groben Visualisierung der Dynamik.

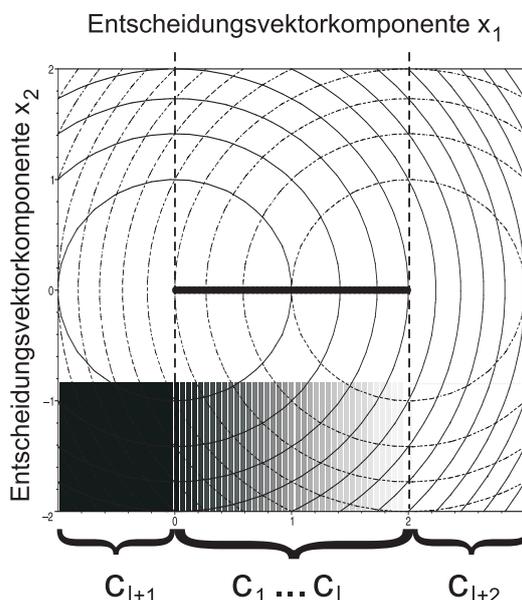


ABBILDUNG 5.2: Schematische Darstellung der Färbung von Lösungen des speziellen Problems  $\mathcal{F}_{\odot}^2$ , um damit eine Visualisierungsmethode für die Lösungsbildung in der Populationsstruktur zu entwerfen.

beider Kriterien. Abbildung 5.3 zeigt die Erstellung einer solchen Populationsvisualisierung beispielhaft. Jedem Individuum auf dem Torus wird entsprechend seiner Ausrichtung bzgl. der beiden Kriterien ein Farbwert der Farbskala zugeordnet. Eine optimale Lösung für Kriterium 1 erhält so den Farbwert  $c_1$ , während die optimale Lösung bzgl. Kriterium 2 den Farbwert  $c_l$  erhält. Alle Kompromisse erhalten abgestufte Farbwerte. Es soll hier noch einmal darauf hingewiesen werden, dass diese Visualisierungsmethode nur für wenige Probleme anwendbar ist, insbesondere für Probleme, deren Pareto-Menge durch eine Gerade darstellbar ist. Weiterhin gilt, dass diese Darstellungsform keinen Aufschluss über Konvergenz von Lösungen gibt. Lediglich die Nähe einer Lösung zu den konfliktären Kriterien kann damit abgeschätzt werden.

### 5.2.2 Analyse des Mutationseinflusses

Für eine erste Betrachtung wird das Räuber-Beute-Modell exemplarisch mit zwei Räubern auf einem  $40 \times 40$ -Torus instanziiert. Um das Problem  $\mathcal{F}_{\odot}^2$  zu lösen, wird weiterhin ein Räuber mit dem Kriterium  $f_1(\vec{x}) = x_1^2 + x_2^2$  und ein zweiter Räuber mit dem Kriterium  $f_2(\vec{x}) = (x_1 - 2)^2 + x_2^2$  zur jeweils einkriteriellen Selektion ausgestattet. Schließlich werden beide Räuber mit normalverteilter Mutation ( $\sigma = 0.1$ , siehe Gleichung (2.3)) als einzigen Operator ausgestattet. Beide Räuber bewegen sich gleichverteilt mit einschrittigem Zufallslauf und führen 100.000 Funktionsauswertungen durch. Abbildung 5.4 (links) zeigt das Ergebnis eines solchen Laufes in Form der approximierten Pareto-Menge.

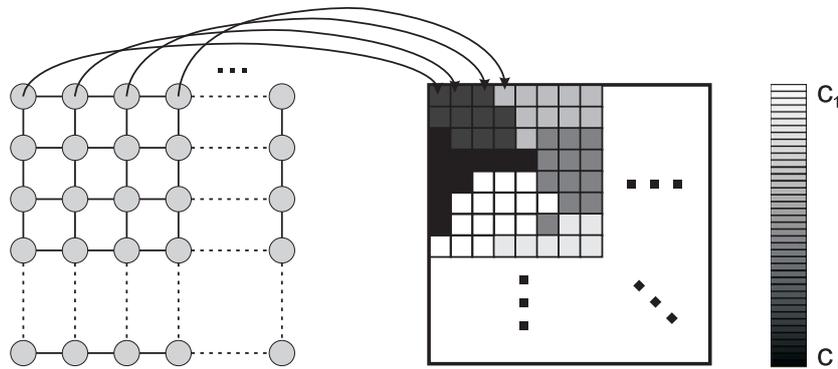


ABBILDUNG 5.3: Beispielhafte Darstellung der Visualisierung von Lösungsqualität in der betrachteten strukturierten Population. Den fest auf dem Torus angeordneten Lösungen (links) werden entsprechend ihrer Lösungsgüte festgelegte Farbwerte in Teilquadranten (rechts) zugeordnet. Die Farbskala gibt die diskrete Unterteilung des betrachteten Lösungsintervalls vor.

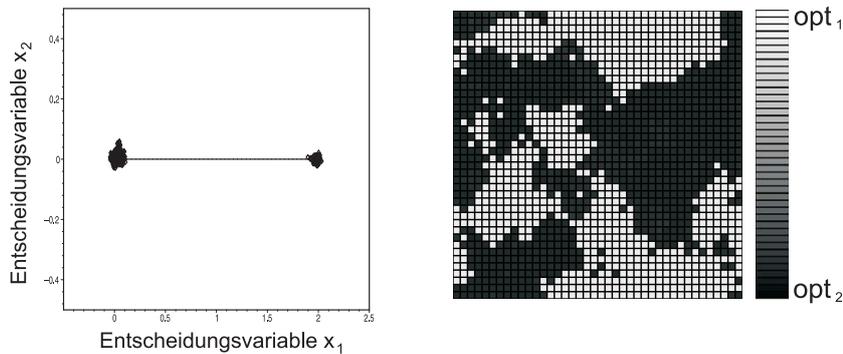


ABBILDUNG 5.4: Approximierte Paretomenge (links) und zugehöriges Populationsabbild (rechts) nach 100.000 Funktionsauswertungen auf dem Problem  $\mathcal{F}_2^{\otimes}$  mit Mutation als alleiniger Variation. Links ist die ermittelte Lösungsmenge dargestellt, die sich ausschließlich auf die Extrempunkte der analytisch bestimmten Paretomenge konzentriert. Rechts ist die entsprechend der Lösungsqualität eingefärbte Beutepopulation auf dem Torus dargestellt.

Die Betrachtung zeigt, dass die Lösungsmenge in zwei Teile zerfällt, die sich jeweils um die Extremwerte der analytisch bestimmten Paretomenge anordnen. Bereits Laumanns und Kollegen haben vermutet, dass sich stabile Lösungsmengen bilden, aber erst die zuvor eingeführte Visualisierungsmethode für die Population ermöglicht es nun, diese Annahme zu bestätigen. Wie der Abbildung 5.4 (rechts) zu entnehmen ist, haben sich zusammenhängende Gebiete auf der toroidalen Populationsstruktur gebildet, die ausschließlich eine der Extremlösungen repräsentieren. Interessant ist zudem die Beobachtung, dass die Grenzen zwischen den Gebieten sehr scharf sind. Es gibt – und dies ist nach der Kenntnis der approximierten Paretomenge nicht überraschend – keine Individuen, die einen Kompromiss bezüglich beider Kriterien, insbesondere bezüglich der  $x_1$ -Komponente darstellen. Es ist also offensichtlich, dass das Räuber-Beute-Modell bereits bei einer sehr einfachen Problemstellung an seine Grenzen stößt und sich der gewünschte Emergenzeffekt

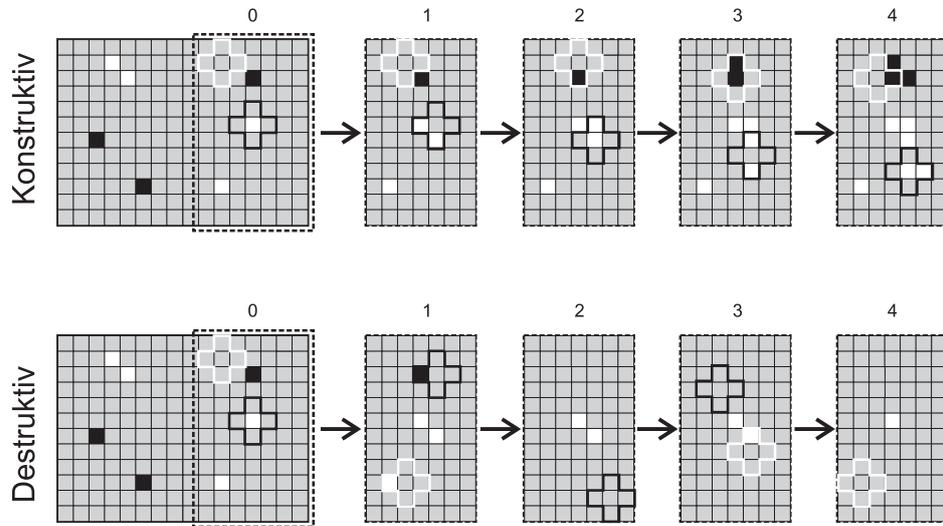


ABBILDUNG 5.5: Schematische Darstellung der Ausbildung und Auflösung von festen Nachbarschaften mit Lösungen ähnlicher Qualität durch kleinschrittige Räuberbewegungen (konstruktiv, oben) und zufällige Sprungbewegungen des Räubers (destruktiv, unten).

zur Bildung von Kompromissen nicht einstellt. Kompromisslösungen überleben die Anwendung von einkriterieller Selektion und ausschließlicher Mutation offenbar nicht. Um spätere Veränderungen am Modell vorschlagen zu können, muss dieser Effekt genauer untersucht werden. Dabei sind wir insbesondere daran interessiert qualitativ zu begründen, ob und warum eine Population zwangsläufig in die beobachtete Situation konvergiert und dort verbleibt.

Wir bauen unsere Argumentation nun schrittweise auf. Zuerst setzen wir voraus, dass der vom Räuber bezüglich seines Kriteriums induzierte evolutionäre Prozess im Allgemeinen eine Annäherung an das Optimum gewährleistet.<sup>6</sup> Dadurch ist sicher gestellt, dass über die Zeit innerhalb der Population Lösungen entstehen können, die nahe am Optimum des jeweiligen Kriteriums liegen. Zur Vereinfachung nehmen wir an, dass diese Lösungen – entsprechend unserem Farbschema – bei hinreichender Nähe zum Optimum von Kriterium 1 eine weiße und bei ausreichender Nähe zum Optimum von Kriterium 2 eine schwarze Färbung erhalten.<sup>7</sup> Alle übrigen Lösungen erhalten eine dritte, graue Farbe, siehe beispielhaft Abbildung 5.5. Das zu Beginn durchgeführte Experiment entspricht der konstruktiven Betrachtungsweise und kann folgendermaßen beschrieben werden: Der Räuber führt jeweils einen zufälligen Schritt auf dem Populationsgitter aus und spannt eine Selektions- und Rekombinationsnachbarschaft auf. Befindet sich, wie hier beispielhaft in Schritt 0, eine dem Optimum des Räuber Kriteriums nahe Lösung in der Nachbarschaft, sorgt der Reproduktionsmechanismus mit großer Wahrchein-

<sup>6</sup>Wir beziehen uns hier auf den evolutionären Pfad, der durch die Selektion beschriftet wird. Eine Aussage zum speziellen Konvergenzverhalten steht nicht im Fokus der Betrachtung.

<sup>7</sup>Die zugehörigen Räuber und ihre Nachbarschaftsgrenzen erhalten die entsprechenden Komplementärfarben, um die Lesbarkeit zu verbessern.

lichkeit für die Erstellung einer weiteren „guten“ Lösung. Dies passiert, da das schlechteste Individuum gemäß Algorithmus 1 zum Löschen markiert wird (dies ist sicher nicht die weiße Lösung) und schließlich eine bessere Lösung zur Reproduktion durch Mutation verwendet wird. Ist die Schrittweite der Mutation hinreichend klein, wird sich das neue Individuum in den meisten Fällen nicht stark von seinem Elternteil unterscheiden. Bewegt sich nun der Räuber und damit die Selektionsnachbarschaft nur geringfügig weiter, so liegt oft im nächsten Schritt wieder ein „gutes“ zuvor erzeugtes Individuum in der Nachbarschaft. Mittelfristig führt dies dazu, dass der Räuber gleichsam eine Spur der Verbesserung auf dem Torus entstehen lässt. Wiederholtes Besuchen von Positionen kann schließlich dazu führen, dass ein größeres Gebiet ähnlicher Lösungen entsteht. Die Beschreibung trifft im gleichen Maße auf den Räuber zu, der Individuen bzgl. des schwarzen Optimums bevorzugt.

Im Allgemeinen kann die Räuberaktivität jedoch auch destruktiv sein, nämlich dann, wenn ein Räuber auf eine für einen anderen Räuber (also bezüglich des anderen Kriteriums) „gute“ Lösung trifft. In diesem Fall ist die Lösung für den gerade betrachteten Räuber „schlecht“ und fällt diesem noch vor den grauen Lösungen zum Opfer. Dieser Effekt wird verstärkt, wenn sich der Räuber mit großen Schritten über den Torus bewegt, also die Lokalität der Aktion aufgehoben wird. Dies kann in verschiedenen Situationen der Fall sein:

1. Der Räuber hat eine große Zufallslaufschrittweite. Dadurch entsteht eine annähernd gleiche Wahrscheinlichkeit für jede Position des Torus als nächstes erreicht zu werden. So wird aber auch verhindert, dass ein Räuber eine Spur „guter“ Individuen und damit mittelfristig stabile Gebiete auf dem Torus ausbilden kann, siehe die schematische Darstellung in Abbildung 5.5 (unten).
2. Der Torus ist mit einer großen Zahl von Räubern bestückt. In diesem Falle arbeiten oftmals Räuber unterschiedlicher Art im gleichen lokalen Gebiet. Dies ist dann annähernd äquivalent zu dem obigen Fall, wo jede Position von jedem Räuber mit gleicher Wahrscheinlichkeit erreicht werden kann und führt zu demselben, destruktiven Effekt, siehe beispielhaft Abbildung 4.2 (rechts).

Insgesamt deckt sich unsere Modellvorstellung auch mit den von Laumanns et al. (1998) beobachteten Effekten. In seiner Diplomarbeit hat Laumanns (1999) genau diese Effekte bei der Erhöhung der Räuberanzahl und der Vergrößerung der Zufallslaufschrittweite beobachtet. Ai (2005) hat diese Beobachtungen 2005 experimentell bestätigt. Auch der in Abbildung 4.2 dargestellte Effekt durch die Vergrößerung der Selektions- und Reproduktionsnachbarschaft lässt sich durch die gleiche Argumentation in unserem Modell erklären.

Ausgehend von den bisherigen Überlegungen betrachten wir die Dynamik innerhalb und an den Grenzen der entstandenen Gebiete nun formal. Dabei beschränken wir uns explizit auf eine Abschätzung des Verhaltens, eben soweit, wie wir es unter dem Einfluss der Zufallskomponenten handhaben können. Dies erlaubt

uns schließlich qualitativ zu beurteilen, inwieweit die Ausbildung von Kompromisslösungen im Rahmen dieses Modells überhaupt möglich ist. Vor der Durchführung der Analyse treffen wir die folgenden – teilweise etwas vereinfachenden – Modellannahmen und beschreiben die algorithmischen Rahmenbedingungen für die Durchführung.

**Zweikriterielles Problem:** Wie für die vorherige Betrachtung und die Visualisierungsmethode, nutzen wir für die Analyse das einfache zweikriterielle Problem  $\mathcal{F}_{\odot}^2$  aus Beispiel 1.11. Wir wiederholen hier die Definition

$$\mathcal{F}_{\odot}^m : \mathbb{R}^n \rightarrow \mathbb{R}^m \text{ mit } \vec{x} \in \mathbb{R}^n, m, n \in \mathbb{N}$$

$$\mathcal{F}_{\odot}^m(\vec{x}) = \begin{pmatrix} (\vec{x} - \vec{c}_1)^2 \\ \vdots \\ (\vec{x} - \vec{c}_m)^2 \end{pmatrix}, \vec{c}_i \in \mathbb{R}^n, \text{ konstant}$$

$$\vec{c}_i \neq \vec{c}_j \text{ für } i \neq j$$

und formulieren es für unsere Betrachtung als zweikriterielles Problem mit  $\vec{c}_1 = (0, 0)^T$  und  $\vec{c}_2 = (\tau, 0)^T$ ,  $0 < \tau \in \mathbb{R}$ . So ergibt sich:

$$\mathcal{F}_{\odot}^2(\vec{x}) = \begin{pmatrix} x_1^2 + x_2^2 \\ (x_1 - \tau)^2 + x_2^2 \end{pmatrix}$$

Analog zur Visualisierungsmethode aus Abschnitt 5.2.1 nehmen wir gute Lösungen für Kriterium 1 als weiß und gute Lösungen für Kriterium 2 als schwarz an.

**Räuber und Nachbarschaft:** Wir führen die Analyse aus der Sicht eines einzelnen Räubers durch. Für diesen Räuber nehmen wir an, dass er stationär ist, also auf einer einzigen Position im Torus verweilt. Um die Position des Räubers herum wird eine Selektions- und Rekombinationsnachbarschaft mit dem Radius  $r$  aufgespannt, so dass diese Nachbarschaft entsprechend Vorschrift (4.1) genau  $(r + 1)^2 + r^2$  Individuen enthält. Für alle Individuen nehmen wir weiterhin an, dass sie sich, wie in der vorherigen Diskussion des konstruktiven Falles dargestellt, bezüglich eines der beiden Kriterien stark spezialisiert haben. Es gilt also, dass die Nachbarschaft ausschließlich weiße und schwarze Individuen enthält. Die Anordnung auf der Gitterstruktur spielt dabei innerhalb der Nachbarschaft keine Rolle. Der Räuber verfolgt das schwarze Kriterium, bewertet also während der Selektion schwarze Individuen als „gut“.

**Selektion und Variation:** In dem durch Algorithmus 1 beschriebenen Räuberverhalten wählt der Räuber aus den in der Nachbarschaft enthaltenen Individuen das bezüglich seines Kriteriums schlechteste Individuum zur Ersetzung aus. Ein

Kandidat für die Ersetzung entsteht durch Variation eines der verbliebenen Individuen der Nachbarschaft. Wir nehmen hier an, dass die Auswahl des Individuums zur Erzeugung des Ersetzungskandidaten *gleichverteilt* aus den  $(r + 1)^2 + r^2 - 1$  restlichen Individuen der Nachbarschaft getroffen wird.

Die eingesetzte Mutation folgt einer von der Mutationsschrittweite  $\sigma$  abhängigen Verteilungsfunktion. Im Kontext evolutionärer Algorithmen kann dies eine Normalverteilung sein, wobei  $\sigma$  dann die Standardabweichung vom Mittelwert 0 angibt. In unserer Analyse interessiert jedoch nur die aus der Verteilungsfunktion resultierende Wahrscheinlichkeit für eine sich durch die Mutation ergebende Qualitätsveränderung eines Individuums. In Anlehnung an die Visualisierungsmethode und die damit verbundene Diskretisierung der Qualitätswerte einer Lösung, bezeichnen wir eine solche Veränderung als *Farbwechsel* eines Individuums. Der Farbwechsel bedeutet, dass ein Nachkomme, der durch die Mutation eines Elternteils entsteht, eine vom Elternteil verschiedene Farbe, also eine andere Qualität bezüglich der betrachteten Kriterien annimmt. Vereinfacht und für unsere Betrachtung ausreichend nehmen wir an, dass dieser Farbwechsel nur von zwei Farben – nämlich von schwarz zu weiß – angezeigt wird. Eine Abstufung wie in unserer Visualisierung nehmen wir nicht vor. Wir bezeichnen  $\epsilon$  als obere Schranke der Wahrscheinlichkeit für das Beibehalten der Elternfarbe durch den Nachkommen. Damit ergibt sich  $\bar{\epsilon} = (1 - \epsilon)$  als untere Schranke der Wahrscheinlichkeit für einen Farbwechsel durch Mutation<sup>8</sup>.

Vereinfacht betrachten wir für den spontanen Farbwechsel eines Individuums nur die  $x_1$ -Komponente des oben festgelegten Problems  $\mathcal{F}_{\odot}^2$  zweigeteilt als

$$C, C' : \mathbb{R} \rightarrow \{\text{schwarz, weiß}\}$$

mit

$$C(x_1) = \begin{cases} \text{weiß,} & \text{wenn } x_1 < \tau \\ \text{schwarz,} & \text{wenn } x_1 \geq \tau \end{cases}$$

für die Ausgangssituation eines weißen Individuums und

$$C'(x_1) = \begin{cases} \text{schwarz,} & \text{wenn } x_1 > 0 \\ \text{weiß,} & \text{wenn } x_1 \leq 0 \end{cases}$$

für den umgekehrten Ausgangsfall eines schwarzen Individuums. Damit entsteht eine neutrale Zone bezüglich der  $x_1$ -Komponente zwischen den beiden Extremlösungen, in der kein Farbwechsel stattfindet und die genau unsere vereinfachende Diskretisierung in zwei Farben ermöglicht, siehe Abbildung 5.6.

<sup>8</sup>Der Wert  $\bar{\epsilon} = 1 - \epsilon$  könnte etwa die untere Schranke der Wahrscheinlichkeit angeben, dass ausgehend von einem bezüglich Kriterium 1 optimalen Elternindividuum eine Mutation derart stattfindet, dass der resultierende Nachkomme bezüglich Kriterium 2 optimal ist. Die Wahrscheinlichkeit  $\epsilon$  würde dann in der Umkehrung eine obere Schranke für das Entstehen eines dem Elternindividuum ähnlichen Nachkommen darstellen.

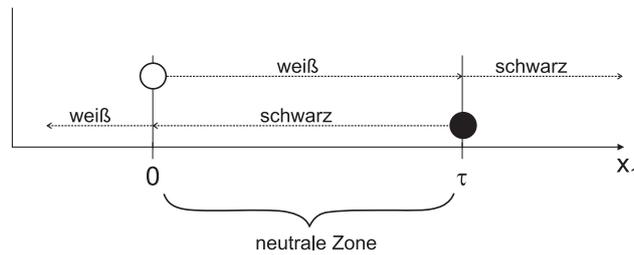


ABBILDUNG 5.6: Schematische Darstellung der eingesetzten diskretisierten Farbgebung. Eine weiße Lösung wird nach Mutation solange als weiß angesehen, bis sie den Extremwert  $\tau$  in der  $x_1$ -Komponente des Entscheidungsraums erreicht bzw. überschritten hat. Ebenso wird eine schwarze Lösung nach Mutation weiterhin als schwarz betrachtet, solange sie die Extremlösung 0 in der  $x_1$ -Komponente nicht erreicht bzw. überschritten hat. Damit entsteht eine neutrale Zone zwischen den Extremösungen.

**Stochastischer Prozess:** Mittels der vorherigen Festlegungen können wir die Entwicklung der Individuen in der Nachbarschaft des stationären Räubers durch einen stochastischen Prozess in Form einer Markow-Kette darstellen. Entsprechend der Nachbarschaftsgröße definieren wir  $(r+1)^2+r^2+1$  Zustände für die Ausprägung der Nachbarschaft. Dabei beschreibt  $s_0$  eine Nachbarschaft, die ausschließlich aus weißen Individuen besteht. Der Zustand  $s_{(r+1)^2+r^2}$  bezeichnet eine Nachbarschaft bestehend aus ausschließlich schwarzen Individuen, siehe Abbildung 5.7. Die übrigen Zustände bezeichnen Nachbarschaften mit (von  $s_1$  bis  $s_{(r+1)^2+r^2-1}$ ) zunehmend vielen schwarzen Individuen.

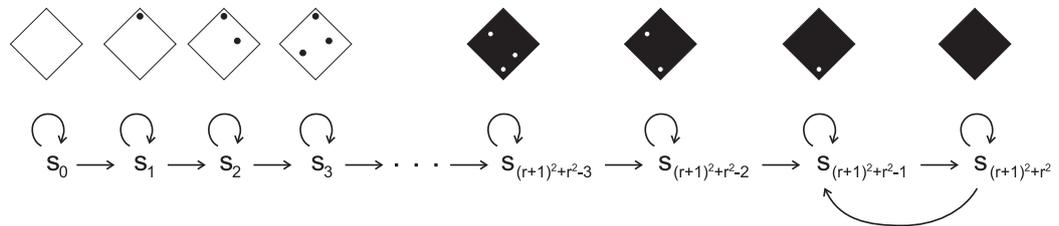


ABBILDUNG 5.7: Darstellung der Markow-Kette bestehend aus  $(r+1)^2+r^2+1$  Zuständen und gerichteten Übergängen zur Beschreibung des Mutationseinflusses in einer Nachbarschaft mit Nachbarschaftsradius  $r$ . Zusätzlich zu den Zuständen  $s_0$  bis  $s_{(r+1)^2+r^2}$  sind schematische Darstellungen der Nachbarschaftskonfiguration gegeben. Der Zustand  $s_0$  beschreibt eine Nachbarschaft mit ausschließlich weißen Individuen, in  $s_1$  befindet sich ein schwarzes Individuum in der Nachbarschaft usw., bis in  $s_{(r+1)^2+r^2}$  alle Individuen der Nachbarschaft schwarz sind.

Die Transitionen zwischen den Zuständen der Markow-Kette ergeben sich direkt aus dem Räuberverhalten und der Wahrscheinlichkeit  $\bar{\epsilon} = (1 - \epsilon)$  für einen Farbwechsel. Allgemein beschreibt  $P_{s_i \rightarrow s_j}$  die Übergangswahrscheinlichkeit von  $s_i$  zu  $s_j$  mit  $i, j \in \{0, \dots, (r+1)^2+r^2\}$ . Für den ersten Zustand  $s_0$  ergibt sich so:

$$\begin{aligned} P_{s_0 \rightarrow s_0} &= \epsilon \\ P_{s_0 \rightarrow s_1} &= \bar{\epsilon} = 1 - \epsilon \end{aligned}$$

Allgemein ergeben sich die Übergangswahrscheinlichkeiten mit  $i = 0, \dots, (r+1)^2 + r^2 - 1$  und  $j = i + 1$  als:

$$\begin{aligned} P_{s_i \rightarrow s_i} &= \frac{(r+1)^2 + r^2 - (1+i)}{(r+1)^2 + r^2 - 1} \epsilon + \frac{i}{(r+1)^2 + r^2 - 1} \bar{\epsilon} \\ &= 1 - \frac{i}{(r+1)^2 + r^2 - 1} - \bar{\epsilon} \left( \frac{(r+1)^2 + r^2 - 2i - 1}{(r+1)^2 + r^2 - 1} \right) \\ &= \epsilon - i \frac{1 - 2\bar{\epsilon}}{(r+1)^2 + r^2 - 1} \end{aligned}$$

$$\begin{aligned} P_{s_i \rightarrow s_{i+1}} &= 1 - P_{s_i \rightarrow s_i} \\ &= \frac{i}{(r+1)^2 + r^2 - 1} + \bar{\epsilon} \left( \frac{(r+1)^2 + r^2 - 2i - 1}{(r+1)^2 + r^2 - 1} \right) \\ &= \bar{\epsilon} - i \frac{1 - 2\bar{\epsilon}}{(r+1)^2 + r^2 - 1} \end{aligned}$$

In jedem der Zustände gibt es ausschließlich zwei mögliche Übergänge: der Übergang in denselben Zustand oder der Übergang in den nächst höheren Zustand. Dabei beschreibt der nächst höhere Zustand eine Nachbarschaft mit einem zusätzlichen schwarzen Individuum. Durch die grundsätzliche Ersetzung des schlechtesten Individuums, ist ein Rückschritt in der Markow-Kette nicht möglich. Da der Räuber in unserem Modell schwarze Individuen als gut bewertet, wird, solange ein weißes Individuum in der Nachbarschaft vorhanden ist, ein weißes Individuum zur Ersetzung ausgewählt. Der Nachkomme wird dann aus einem zufällig gewählten übrigen Individuum aus der Nachbarschaft erzeugt. Dabei entsteht entweder erneut ein weißes Individuum (Verweilen im selben Zustand) oder ein schwarzes Individuum (Übergang in den nächsthöheren Zustand). Der erste Fall tritt ein, wenn das zufällig gewählte Elternindividuum weiß ist und eine geringe Mutation zu einer nur geringen Veränderung im Phänotyp des Nachkommens führt, oder wenn aus einem schwarzen Elternindividuum durch eine starke Mutation ein weißer Nachkomme erzeugt wird. Ein schwarzes Nachkommenindividuum entsteht, wenn ein schwarzes Elternindividuum wenig verändert, oder ein weißes Elternindividuum zu einem Farbwechsel hin stark verändert wird.

Für den letzten Zustand ergibt sich ein Sonderfall. Hier liegt eine vollständig schwarze Population vor. In dieser Situation ist ein Rücksprung in den vorherigen Zustand (ein weißes Individuum) möglich, wenn eine starke Mutation zu einem Farbwechsel führt.

$$\begin{aligned} P_{s_{(r+1)^2+r^2} \rightarrow s_{(r+1)^2+r^2}} &= \epsilon \\ P_{s_{(r+1)^2+r^2} \rightarrow s_{(r+1)^2+r^2-1}} &= \bar{\epsilon} \end{aligned}$$

Mit Hilfe der Formulierung der Übergangswahrscheinlichkeiten können wir die Übergangsmatrix  $\mathbf{T}$  für die Markow-Kette aus Abbildung 5.7 erzeugen, siehe Gleichung (5.1). In der Matrix sind die Übergangswahrscheinlichkeiten aufsteigend nach dem Zustandsindex in Zeilen bzw. Spalten angeordnet.

$$\mathbf{T} = \begin{bmatrix} \epsilon & \bar{\epsilon} & 0 & 0 & \dots & 0 \\ 0 & P_{s_1 \rightarrow s_1} & P_{s_1 \rightarrow s_2} & 0 & & 0 \\ 0 & 0 & & \ddots & & \vdots \\ \vdots & & & & P_{s_i \rightarrow s_i} & P_{s_i \rightarrow s_{i+1}} & 0 \\ & & & & & \ddots & \ddots \\ 0 & 0 & & & & 0 & \bar{\epsilon} & \epsilon \\ 0 & 0 & \dots & & & 0 & \bar{\epsilon} & \epsilon \end{bmatrix} \quad (5.1)$$

Für jeden Anfangszustand kann diejenige Wahrscheinlichkeitsverteilung, die angibt in welchem Zustand der Markow-Kette man sich nach  $k$  Schritten befindet, mit Hilfe die Übergangsmatrix  $\mathbf{T}$  bestimmt werden. Der Verteilungsvektor für jeden Anfangszustand ist im zugehörigen Zeilenvektor der Matrix  $\mathbf{T}^k$  zu finden.<sup>9</sup>

Die Betrachtung des Graphen der Markow-Kette zeigt, dass das erstellte Modell offenbar zwangsläufig in die beiden letzten Zustände tendiert. Solange für die ersten  $(r+1)^2 + r^2 - 2$  Zustände jeweils ein Übergang in den nächst höheren Zustand mit einer Wahrscheinlichkeit größer Null existiert und keine Rückschritte in vorherige Zustände möglich sind, konvergiert die Markow-Kette zwangsläufig hin zu den beiden letzten Zuständen. Diese Zustände beschreiben nach Motwani & Raghavan (1995) eine starke Komponente des Übergangsgraphen. Für diese Matrix lässt sich die sogenannte stationäre Verteilung über die Konvergenz der Folge  $\mathbf{T}^k$  mit  $k \rightarrow \infty$  ermitteln, also jene Verteilung, in die das System unabhängig von der Startbedingung konvergiert. Bei der Ermittlung des stationären Verteilungsvektors  $\vec{q}$  hilft, dass dieser der Eigenvektor der transponierten Matrix  $\mathbf{T}^T$  ist (Motwani & Raghavan, 1995; Storch & Wiebe, 1999):

$$\mathbf{T}^T \vec{q} = \vec{q}.$$

Somit kann er durch Lösen des aus

$$(\mathbf{T}^T - \mathbf{1})\vec{q} = 0$$

entstehenden homogenen Gleichungssystems bestimmt werden. Dabei ist  $\mathbf{1}$  die Einheitsmatrix. Hilfreich ist für die Auflösung des Gleichungssystems, dass für den

<sup>9</sup>Es findet sich also in Zeile 1 der Matrix die Wahrscheinlichkeitsverteilung für den Aufenthalt in allen Zuständen nach  $k$  Schritten, ausgehend von Zustand  $s_0$ . Analog finden sich die Verteilungen für den Anfangszustand  $s_1$  in Zeile 2, für den Anfangszustand  $s_2$  in Zeile 3 usw.

Zustandswahrscheinlichkeitsvektor  $\vec{q}$  zusätzlich  $\sum_{i=1}^{(r+1)^2+r^2+1} q_i = 1$  gilt. So erhalten wir für den Verteilungsvektor  $\vec{q}$  unter der Markow-Kette mit Übergangsmatrix  $\mathbf{T}$  folgendes Ergebnis:

$$\vec{q} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 - \epsilon \\ \epsilon \end{pmatrix} \quad (5.2)$$

Übertragen wir diese allgemeine Betrachtung auf die im Räuber-Beute-Modell verwendete Normalverteilung für die von der Mutation eingebrachte Störung des Elternindividuums, so hängt  $\epsilon$  direkt von der Mutationsschrittweite  $\sigma$  ab. Wird  $\sigma$  klein gewählt, so ist die Wahrscheinlichkeit für einen großen Unterschied zwischen Eltern- und Nachkommenindividuum ebenfalls klein. Das bedeutet, die Wahrscheinlichkeit für einen Farbwechsel durch Mutation ist gering. Wird die Mutationsschrittweite hingegen groß gewählt, so ist auch die Wahrscheinlichkeit für einen Farbwechsel groß. Gemäß des stationären Verteilungsvektors  $\vec{q}$  konvergiert das Modell jedoch in jedem Fall in die beiden letzten Zustände, egal, in welchem Zustand gestartet wird. Dies führt zu:

**Folgerung 5.1 (Extremalentwicklung der Nachbarschaft)**

*Unabhängig von der Mutationsschrittweite und unter einkriterieller Selektion entsteht durch den Einfluss eines stationären Räubers eine (annähernd) vollständig bezüglich des Räuber criteriums ausgeprägte Nachbarschaft.*

Die Dynamik des Räubers im Räuber-Beute-Modell impliziert im Kontext des Markow-Prozesses jedoch  $k \ll \infty$ . So prägt sich nicht direkt eine stabile Nachbarschaft aus. Dennoch trifft der Räuber für jede aufgespannte Nachbarschaft um seine aktuelle Position herum auf einen Zustand der Markow-Kette und kann diese Nachbarschaft ausschließlich in einen gleich- oder höherwertigen Zustand überführen. Gehen wir zugleich von einer geringen Zufallslaufschrittweite des Räubers auf dem Torus aus (wir haben für unser Modell bisher immer eine Zufallsbewegung der Länge 1 angenommen), so ergibt sich nach der Räuberbewegung eine zur betrachteten Nachbarschaft ähnliche Nachbarschaft. Selbst im Falle  $r = 1$  ist mindestens ein Individuum der vorherigen Nachbarschaft enthalten. Diese starke Lokalität und die nach wie vor geltende Extremalentwicklung führt zu der Erhaltung von stabilen Gebieten. Diese folgen genau wie im konstruktiven Fall von Abbildung 5.5 beschrieben in ihrer Entstehung dem Räuberpfad. Aus der Markow-Kette können wir für diesen Vorgang das Folgende schließen:

**Folgerung 5.2 (Austausch von Lösungen durch Mutation)**

*In einer Population, die unter dem Einfluss verschiedener Räuber stabile Gebiete bezüglich der verfolgten Einzelkriterien ausgebildet hat, findet durch einkriterielle Selektion und anschließende Mutation an den Rändern der Gebiete ein Austausch von Lösungen, jedoch keine Kompromissbildung statt.*

Innerhalb eines voll ausgeprägten Gebietes gilt unser Modell ebenfalls. Hält sich ein Räuber, der schwarze Individuen als gut bewertet, in einer vollständig schwarzen Nachbarschaft auf, bleibt diese Nachbarschaft schwarz. Interessanterweise impliziert das Modell bei kleiner Mutationsschrittweite und damit geringer Übergangswahrscheinlichkeit in den nächst höheren Zustand, zusammen mit der durch die Dynamik des Räubers hervorgerufenen kurzen Verweildauer in der Nachbarschaft eine gewisse „Immunität“ der Nachbarschaft gegen das Eindringen eines komplementären Räubers.

**Folgerung 5.3 (Unverletzlichkeit der Nachbarschaft)**

*Bei hinreichend kleiner Mutationsschrittweite und einkriterieller Selektion wird eine auf ein Kriterium angepasste Nachbarschaft zu einem stabilen Gebiet, welches gegen das kurzzeitige Eindringen eines komplementären Räubers weitgehend unempfindlich ist.*

Die drei Folgerungen legen nahe, dass eine Anwendung des Räuber-Beute-Modells unter ausschließlichem und ungerichtetem Mutationseinfluss nicht zu einer Approximation von Kompromisslösungen geeignet ist. Eine ausschließliche Veränderung der Mutationsschrittweite durch Anpassungsmechanismen ist ebenfalls ausgeschlossen, da die Verkleinerung der Schrittweite zur Stabilisierung der entwickelten Gebiete beiträgt und eine Erhöhung der Mutationsschrittweite die Konvergenz in die Extrema vielleicht verzögern aber nicht verhindern kann. Untersuchungen von Grimme (2005) und Grimme & Schmitt (2006) haben gezeigt, dass die Anpassung der Schrittweite<sup>10</sup> zu einer Steigerung des Konvergenzverhaltens in die Extremlösungen der Paretomenge führt und die Stabilisierung der festen Gebiete in der Population verstärkt.

**5.2.3 Analyse des Rekombinationseinflusses**

Als zweiter separater Einfluss auf die Lösungsbildung soll nun der Rekombinationsoperator betrachtet werden. Im Allgemeinen beschreibt die Rekombination wie in Abschnitt 2.2.2 bereits dargestellt eine konvexe Operation, die existierende genetische Information neu kombiniert, jedoch außerhalb der konvexen Hülle der bisherigen Lösungen keine Innovation beisteuert. Viele Autoren berichten, dass ein abschließender Nachweis des Nutzens der Rekombination (oder Crossover bei

<sup>10</sup>Für den speziellen Fall der lokalen Steady-State-Selektion des Räuber-Beute-Modells nutzten die Autoren den von Runarsson vorgeschlagenen Anpassungsmechanismus für asynchrone parallele Evolutionsstrategien (Runarsson, 2003). Eine ausführliche Darstellung des Mechanismus und seiner Übertragung findet sich in den entsprechenden Arbeiten und ist hier nicht Gegenstand der Untersuchung.

GAs) bisher nicht geführt wurde (Bäck, 1996; Beyer, 2001). Dennoch gibt es zwei grundsätzlich verschiedene Hypothesen über den Beitrag der Rekombination zur Lösungsbildung: die Building Block-Hypothese nach Goldberg (1989) und die Genetic Repair-Hypothese nach Beyer (1995, 1997). Während die erste Hypothese davon ausgeht, dass durch die Rekombination gute Eigenschaften der Eltern (die Bausteine oder Building Blocks) zu einer neuen noch besseren Lösung zusammenfinden, geht die Genetic Repair-Hypothese von dem gegenteiligen Mechanismus aus. Die Rekombination fungiert dort als „statistische Fehlerkorrektur“ und sichert das Überleben gemeinsamer Eigenschaften von Eltern. Diesen gemeinsamen Eigenschaften wird schließlich nach Beyer (2001) der statistisch größte Nutzen zugeordnet.<sup>11</sup>

Im Falle der mehrkriteriellen Optimierung wollen wir nun untersuchen, inwieweit die Rekombination unter bestimmten Bedingungen ebenfalls nützlich für die Lösungsbildung eingesetzt werden kann. Dazu verwenden wir wieder das im vorherigen Abschnitt betrachtete Problem  $\mathcal{F}_{\odot}^2$ , und auch die Visualisierung aus Abschnitt 5.2.1. Außerdem nutzen wir in dieser Betrachtung die von Grimme (2005) vorgeschlagene verallgemeinerte intermediäre Simplex-Rekombination, siehe Vorschrift (2.6) in Abschnitt 2.2.2. Dieser Operator benötigt für eine Rekombination im  $n$ -dimensionalen Raum  $n + 1$  Eltern, um ein Simplex aufzuspannen, in dem gleichverteilt zufällig ein Nachkomme erzeugt wird. Im Gegensatz zur lokal intermediären Rekombination ist die Simplex-Rekombination unabhängig von den Koordinatenachsen des Lösungsraumes.<sup>12</sup> Wie Grimme & Schmitt (2006) darstellen, ermöglicht diese Rekombination eine systematische Annäherung an die Lösungsmenge im Suchraum: Das in der konvexen Hülle der Eltern erzeugte Individuum liegt für konvexe Probleme sicher nicht weiter von der Paretomenge entfernt, als der am weitesten entfernte Elternteil. Für die diskrete und intermediäre Rekombination aufgrund ihrer Ausrichtung an den Koordinatenachsen des Suchraumes kann das nicht garantiert werden. Diese Eigenschaft der Simplex-Rekombination charakterisieren Grimme und Schmitt als die Konvergenz und Lösungsqualität verbessernd. Wir wollen basierend auf Ergebnissen von Grimme et al. (2007, 2011) die Betrachtung auf globale und lokale Effekte ausdehnen und das Verhalten der Population genauer untersuchen.

### Globale Eigenschaften

Der offensichtlich globale Effekt der Rekombination resultiert aus der Erzeugung von Individuen ausschließlich in der konvexen Hülle des von den Eltern aufgespannten Simplex. Unter dem Einfluss der Selektion und der Vernachlässigung

---

<sup>11</sup>Beyer führt aus, dass neben der Innovationsgenerierung durch die Mutation und der Auswahl von guten Lösungen durch die Selektion, der Rekombination die Aufgabe der Konservierung von überlebenden Gemeinsamkeiten zukommt.

<sup>12</sup>Die Ermittlung des zufälligen Punktes innerhalb des Simplex geschieht durch eine Transformation in baryzentrische Koordinaten. Für eine genaue Darstellung siehe (Grimme, 2005) oder (Grimme & Schmitt, 2006).

der Mutation kollabiert damit die initiale Population im Bereich des verfolgten Ziels. Es ist also klar, dass die endgültige Position der Individuen im Suchraum stark von dem Bereich, den die initiale Population abdeckt und den verfolgten Kriterien abhängt, siehe Abbildung 5.8. Befindet sich die angestrebte Lösungs-

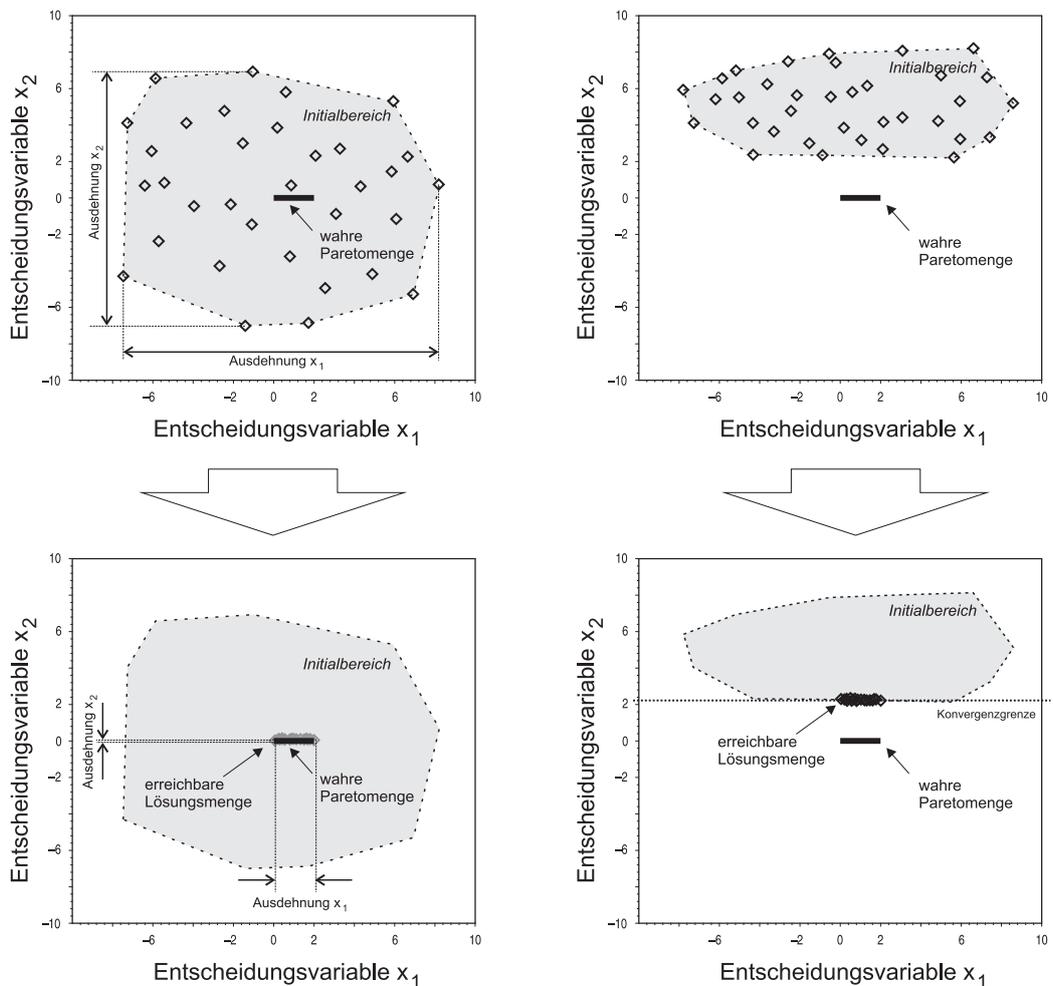


ABBILDUNG 5.8: Darstellung zur Erläuterung der Ausdehnung einer Population für die Bestimmung der Populationsentwicklung unter Rekombinationseinfluss (links) und eine Skizze der Populationsentwicklung bei einem die Paretomenge umschließenden Initialbereich. Die rechte Abbildung stellt schematisch die Begrenzung einer möglichen Konvergenz durch ausschließliche Rekombinationsanwendung dar, wenn die konvexe Hülle der Population die angestrebte Lösungsmenge nicht umschließt. Hier ist eine Konvergenz am Rande der initialen konvexen Hülle zu erwarten.

menge inmitten der konvexen Hülle der initialen Population, so können wir davon ausgehen, dass die Kontraktion der Population unter dem Einfluss von Rekombination und Selektion ihren Schwerpunkt und eine Häufung neuer Individuen in den Bereich der Lösungsmenge verschiebt. Umschließt die initiale Population hingegen

die angestrebte Lösungsmenge nicht, so kann diese durch alleinige Anwendung von Rekombination und Selektion nicht erreicht werden. Unter dem hauptsächlichsten Einfluss der Selektion wird die Population zwar nahe der Grenze der ursprünglichen konvexen Hülle kollabieren, die Grenze kann aufgrund fehlender Mutation (und damit fehlender Innovation in der Entwicklung der Population) nicht überschritten werden.

Unsere weiteren Untersuchungen sollen detaillierter die Entwicklung der Population hin zur angestrebten Lösungsmenge untersuchen. Dazu definieren wir die *maximale Ausdehnung der konvexen Hülle* einer Population als Maß unserer Betrachtung. Die maximale Ausdehnung wird wie in Abbildung 5.8 (links) dargestellt komponentenweise betrachtet und gibt jeweils die größte Distanz zweier Individuen der Population an.

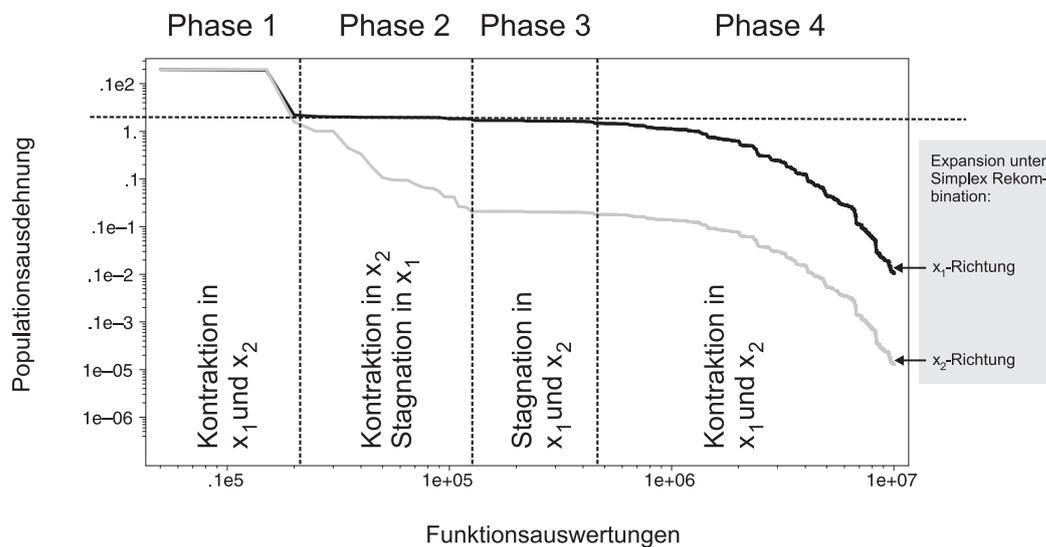


ABBILDUNG 5.9: Exemplarische Darstellung der maximalen Populationsausdehnung (konvexen Hülle) im Suchraum über die Anzahl der Funktionsauswertungen. Der Verlauf kann unter ausschließlichem Einfluss der Simplexrekombination in vier Entwicklungsphasen eingeteilt werden.

Abbildung 5.9 stellt die maximale Populationsausdehnung für die Suchraumkomponenten  $x_1$  und  $x_2$  unseres betrachteten Problems  $\mathcal{F}_{\odot}^2$  in einem Optimierungsprozess über die Anzahl der Funktionsauswertungen dar. In diesem Fall wurde die initiale Population im Bereich  $[-10, 10]^2$  zufällig gebildet, so dass die angestrebte Paretomenge innerhalb der Hülle der Population liegt. Insgesamt können wir über die Zeit vier Phasen in der Entwicklung der Populationsausdehnung feststellen.

**Phase 1, Kooperation und Kontraktion:** Die erste Phase ist durch die Kooperation der Räuber aller Kriterien geprägt. Solange die zur Rekombination verwendeten Individuen teilweise außerhalb des Zielgebietes liegen, ist jeder Fortschritt – egal für welches Kriterium – auch eine positive Weiterentwicklung für das andere Kriterium. Das Zielgebiet definieren wir in unserem Fall als denjenigen Unterraum des gesamten Suchraumes der unbeschränkt in  $x_2$

und begrenzt durch die beiden Extremwerte in  $x_1$  ist. Diese Kooperation führt nach einer gewissen Anlaufphase zu einer Kontraktion der Population.

**Phase 2, Kontraktion und Stagnation:** Die zuvor festgestellte Kontraktion der Population endet für die  $x_1$ -Komponente, wenn das Zielgebiet erreicht ist. Innerhalb dieses Gebietes ist die Entwicklung eines Individuums zugunsten von Kriterium  $f_1$  zugleich eine Verschlechterung bezüglich Kriterium  $f_2$  und umgekehrt. Dies verlangsamt die Kontraktion der Population in diese Richtung (die gestrichelte Linie in Abb. 5.9 markiert eine Ausdehnung von 2, also die Ausdehnung der optimalen Menge im Suchraum bzgl.  $x_1$ ). In der Approximation der  $x_2$ -Komponente besteht jedoch kein Widerspruch, so dass hier eine weitere Kontraktion in Richtung Paretofront zu beobachten ist.

**Phase 3, Stagnation:** In der anschließenden Phase ist eine Stagnation der Ausdehnung in beiden Komponenten zu beobachten, die darin begründet ist, dass die Kriterien für die Selektion zunehmend an Bedeutung verlieren. Durch die Konvexität der Rekombination werden zwangsläufig Individuen im Schwerpunkt der Population gebildet, deren Zahl langsam ansteigt und die in der letzten Phase schließlich die Mehrheit der Population ausmachen.

**Phase 4, Kontraktion:** Mit dieser Phase beginnt erneut eine Kontraktion, in der häufiger Lösungen nahe des Schwerpunktes produziert werden als dies an den Rändern der Population der Fall ist. Die Lösungen im (oder nahe am) Schwerpunkt nehmen wiederum häufiger am Reproduktionsprozess teil und beschleunigen so die Kontraktion der Gesamtpopulation in beiden Komponenten.

### Lokale Eigenschaften

Um insbesondere die Vorgänge in den Übergangsphasen 2 und 3 genauer zu betrachten, beobachten wir erneut die lokale Dynamik innerhalb der Populationsstruktur, diesmal unter ausschließlichem Rekombinationseinfluss. Abbildung 5.10 zeigt beispielhaft die Situation nach der ersten Phase. Viele der Individuen haben das Zielgebiet erreicht und – ähnlich wie im Falle der Mutation – stabile Gebiete um die Extremlösungen herausgebildet. Bei einer Initialisierung der Population kann der Entwicklungsprozess dieser Gebiete ebenfalls mit dem in Abschnitt 5.2.2 eingeführten Modell erklärt werden und folgt insbesondere der Folgerung 5.3 zur Unverletzlichkeit der stabilen Gebiete. Die einkriterielle Selektion bestimmt weitgehend die Entwicklung und die Stabilisierung der Gebiete. Die Abbildung 5.10 zeigt aber auch über den zeitlichen Verlauf ein langsames Verschwinden der Gebiete und einen Übergang zu vorwiegend gemischten Individuen. An dieser Stelle ist die Folgerung 5.2 über den einfachen Austausch von Lösungen an der Grenze von Gebieten nicht mehr anwendbar. Beide Beobachtungen – die Existenz stabiler Gebiete sowie die Vermischung an Grenzen der Population – lassen sich anschaulich mit der schematischen Betrachtung aus Abbildung 5.11 erläutern.

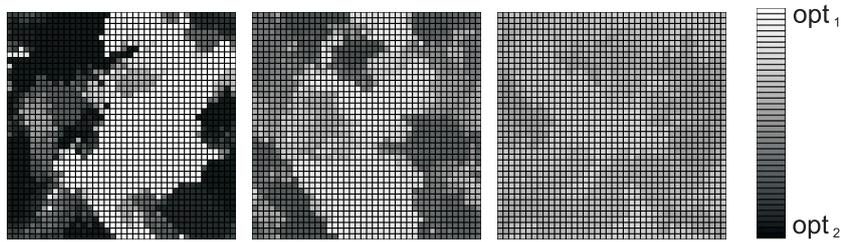


ABBILDUNG 5.10: Beispielhafte Visualisierung der Populationsentwicklung über die Zeit unter ausschließlichem Einfluss der Simplexrekombination. Die Populationsstruktur tendiert zu einer vollständigen Vermischung.

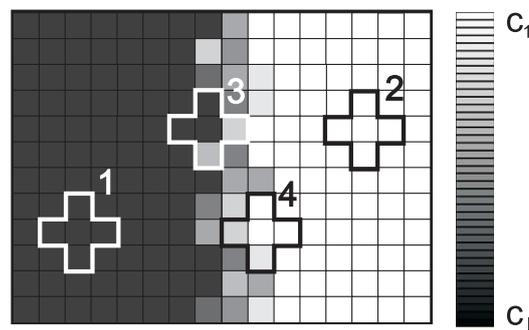


ABBILDUNG 5.11: Schematische Darstellung der Auswirkungen intermediärer Rekombination auf die räumlich verteilte Population. Dabei hat die Rekombination lokal unterschiedlichen Einfluss. In stabilen Gebieten ist der Einfluss zu vernachlässigen, während an den Grenzen zwischen den Gebieten eine Vermischung auftritt.

Insgesamt können in der betrachteten Population mit stabilen Gebieten für jeden Räuber zwei Nachbarschaftskonstellationen entstehen, die in Abbildung 5.11 dargestellt und Gegenstand der folgenden Betrachtung sind:

1. Die Beute in der aufgespannten Nachbarschaft ist vollständig schwarz oder vollständig weiß (markiert mit 1 und 2). In dieser Situation hat die Rekombination annähernd keinen Effekt, da durch ihre Anwendung ein Mischindividuum zweier sehr ähnlicher Elternindividuen erstellt wird, was sicher wieder ähnlich zu den Eltern ist. Dieser Effekt trägt direkt zu einer Angleichung der Individuen und damit zu einer Stabilisierung der Nachbarschaft bei.
2. Der Räuber befindet sich auf der Grenze zweier stabiler Gebiete deren Lösungen unterschiedlichen Kriterien entsprechen. Wenn mehr als eine dem Räuberkriterium widersprechende Lösung in der Nachbarschaft ist, so wird als Nachkomme oftmals ein „gemischtes“ Individuum erzeugt, das für das vorliegende Problem einen Kompromiss darstellt. Dies führt über die Zeit zu einer Zunahme der mittleren Lösungen und zur Etablierung fester Grenzgebiete zwischen den Extremregionen, die ebenfalls gegen Räubereinwirkung stabil sind. Die weitere Zunahme der mittleren Gebiete führt schließlich zu einer vollständigen Kontraktion der Population zum Schwerpunkt der Population.

Die Stagnation in Phase 3 ist so durch die kurzzeitige Stabilisierung der Extremlgebiete und die in Folgerung 5.3 festgehaltene Unverletzbarkeit der Nachbarschaft erklärt. Da sich mittlere Lösungen nur an den Rändern der stabilen Gebiete bilden und nur in einem Bruchteil der Reproduktionen auftreten, verzögert sich die schließlich in Phase 4 beobachtete Kontraktion.

Zusammenfassend können wir aus der Untersuchung der Simplex-Rekombination festhalten, dass sie in unterschiedlichen Phasen der Populationsentwicklung unterschiedliche Beiträge zur Lösungsbildung liefern kann. Durch die ausschließliche Lösungsbildung in der konvexen Hülle der Population kann die Rekombination in der initialen Phase zur Annäherung an die Paretofront beitragen. Angelehnt an den Gedanken der Genetic Repair-Hypothese setzen sich in dieser Phase bei der Rekombination die Gemeinsamkeiten der Lösungen für beide Kriterien durch. In der Phase 2 ist dieser Effekt (wenn auch nur für eine Komponente) weiterhin zu beobachten. Es lässt sich also als Folgerung formulieren:

**Folgerung 5.4 (Kooperation bei der Lösungsbildung)**

*Während der initialen Phase des Evolutionsprozesses im Räuber-Beute-Modell unterstützt die Simplex-Rekombination die Lösungsbildung in kooperativen Bereichen aller Kriterien.*

Aus der Betrachtung der dritten und vierten Phase der Populationsentwicklung und der Untersuchung lokaler Effekte stellen wir für konvexe Probleme fest, dass die Simplex-Rekombination zur Bildung von mittleren Lösungen beiträgt und so Kompromisse erzeugen kann. Wir fassen auch dies zusammen:

**Folgerung 5.5 (Herausbildung von Kompromisslösungen)**

*Die Simplex-Rekombination trägt für konvexe Probleme zu einer Bildung von mittleren Lösungen bei, die schließlich Kompromisse darstellen können.*

Es ist bereits hier anzumerken, dass insbesondere die letzte Folgerung ausschließlich auf die Klasse derjenigen Probleme anzuwenden ist, die über eine konvexe Paretomenge verfügen. Bei nicht konvexen Paretomengen kann der Einsatz von Rekombination sogar zu Lösungsverlusten führen. Wir werden dies in Teil III der Arbeit genauer betrachten.

### 5.2.4 Kombination von Mutation und Rekombination

Die detaillierte Untersuchung der Einflüsse einzelner Variationsoperatoren in den vorherigen Abschnitten hat uns einen Einblick in die Funktionsweise der Mutation und Simplex-Rekombination im Kontext des Räuber-Beute-Modells geliefert. Der einkriterielle Selektionsprozess der Räuber trägt unabhängig vom verwendeten Operator zu einer Herausbildung stabiler Gebiete in der verteilten Population bei. Dies wird bei der Anwendung einer normalverteilten Mutation von einer starken Tendenz hin zu Extremlösungen der betrachteten Kriterien begleitet. Dieser Effekt ist sogar weitgehend unabhängig von der eingestellten Mutationsschrittweite. Die

Betrachtung von Räubern, die die Simplex-Rekombination als Variation einbringen zeigt, dass diese durchaus zur Annäherung an das Zielgebiet und anschließend zur Herstellung von Kompromisslösungen beitragen kann.

Betrachtet man die Eigenschaften der untersuchten Variationsoperatoren gemeinsam, so ergibt sich direkt der Gedanke diese zu kombinieren, um Nachteile der Mutation – nämlich den Verlust von Kompromisslösungen – mit der durch die Rekombination zum Schwerpunkt gerichtete Kontraktion der Population zu kompensieren. In ähnlicher Weise kann der Verlust von Extremlösungen durch die Rekombination von der Mutation ausgeglichen werden. Schließlich, und für die erfolgreiche Anwendung der Rekombination von großer Bedeutung, ist die Fähigkeit der Mutation Innovation in die Population zu bringen.

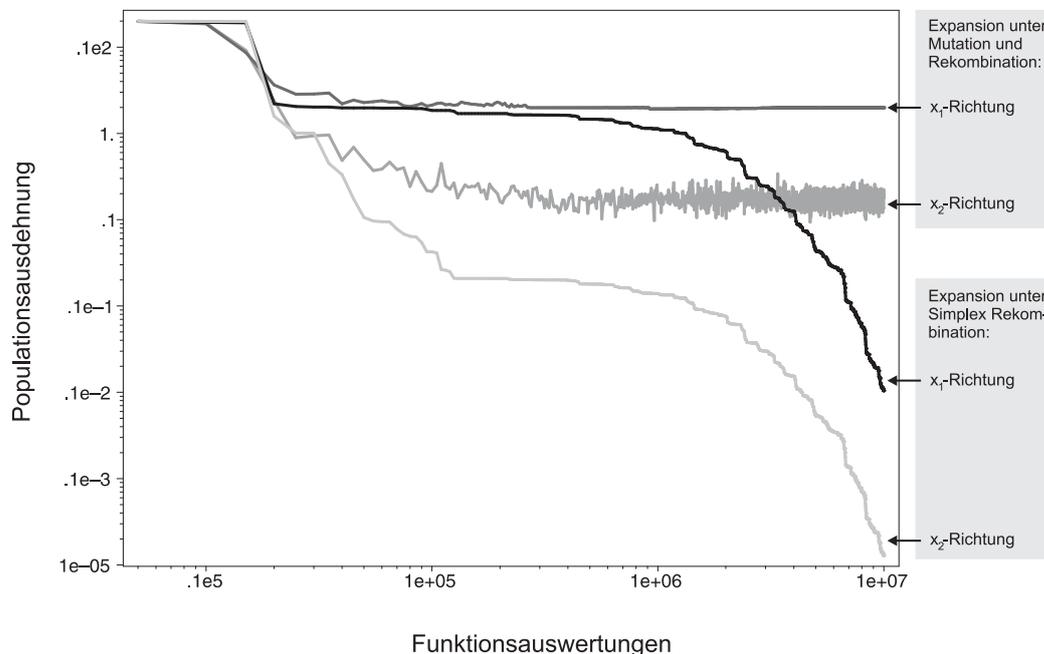


ABBILDUNG 5.12: Erweiterung der Darstellung der maximalen Populationsausdehnung aus Abbildung 5.9 um den Ausdehnungsverlauf für das Szenario von kombinierter Anwendung von Mutation und Rekombination.

Um die gemeinsame Arbeit von Mutation und Rekombination zu betrachten, fügen wir vier Räuber in die Population für das Testproblem  $\mathcal{F}_{\odot}^2$  ein. Jeweils zwei Räuber verfolgen Kriterium  $f_1$  und  $f_2$ , wobei für jedes Kriterium ein Räuber mit Mutation und der andere mit Simplex-Rekombination ausgestattet wird. Wir verwenden weiterhin einen Nachbarschaftsradius von  $\nu = 1$ , eine feste Mutationschrittweite von  $\sigma = 0.1$ , einen  $40 \times 40$ -Torus und  $10^7$  Funktionsauswertungen.

In Abbildung 5.12 stellen wir die Entwicklung der Populationsausdehnung in den Dimensionen  $x_1$  und  $x_2$  dar. Gleichzeitig zeigen wir zum Vergleich die Entwicklung der Rekombination aus Abbildung 5.9. Die Betrachtung zeigt, dass sich in der Entwicklung verschiedene Eigenschaften kombinieren: Zu Beginn spricht die

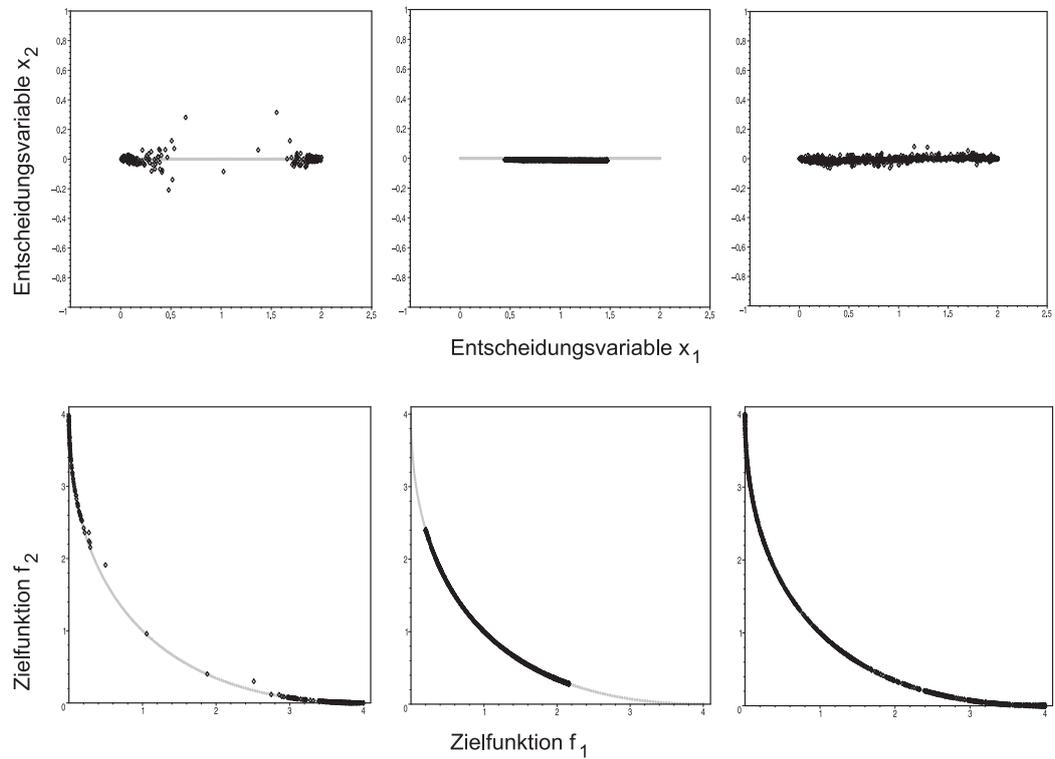


ABBILDUNG 5.13: Darstellung der Auswirkungen einer Kombination von Variationsoperatoren für das Testproblem  $\mathcal{F}_{\otimes}^2$ . Die Extremlösungen (links) werden durch die Mutation und die Kompromisslösungen (Mitte) durch die Rekombination in ausschließlicher Anwendung erzeugt. Eine Kombination beider Operatoren ermöglicht die Approximation der gesamten Paretofront.

schnelle Kontraktion für den Einfluss der Rekombination. Danach sorgt der Einfluss der Mutation dafür, dass die Ausdehnung entsprechend der  $x_1$ -Komponente konstant in der Größe der Ausdehnung der Paretomenge bleibt. Gleichzeitig fällt die Ausdehnung der  $x_2$ -Komponente bis die Größenordnung der eingestellten Mutationsschrittweite erreicht ist. Durch die Anwendung beider Einflüsse wird die Entwicklung der Ausdehnung offenbar stabilisiert.

Die Betrachtung der Paretomenge und der Paretofront in Abbildung 5.13 zeigt, dass die Anwendung beider Variationsoperatoren, eingebracht über verschiedene und unabhängige Räuber, in der Kombination eine gute Approximation der Paretomenge für das Problem ermöglicht. Dies bestätigt insbesondere den in Abschnitt 4.3 angesprochenen Baustein-Ansatz. Dieser erlaubt das separate und wohl-dosierte Einbringen und Kombinieren von Variationseinflüssen über die Räuberindividuen. Wie jedoch das vorliegende Kapitel ebenfalls zeigt, kann dies nur sinnvoll geschehen, wenn die Einflüsse eines jeden Operators im Kontext der Gesamtdynamik des Systems hinreichend verstanden sind. Aufbauend auf dem hier erlangten grundlegenden Verständnis der Modelldynamik können wir nun das Räuber-Beute-

Modell in seiner Leistungsfähigkeit und auch bezüglich seiner Grenzen beurteilen und Erweiterungen vorschlagen.



## **Teil III**

# **Evaluation von Erweiterungen des Räuber-Beute-Modells**



## Kapitel 6

# Die Möglichkeiten und Grenzen des Räuber-Beute-Modells: Eine Vorstudie und Kritik

Die im vorherigen Kapitel durchgeführte Analyse des Einflusses verschiedener Variationsoperatoren auf die Entwicklung der Beutepopulation, haben zu einem genaueren Verständnis der internen Funktionsweisen des Räuber-Beute-Modells geführt. Wir sind nun in der Lage die Auswirkungen von Mutation und Rekombination im jeweiligen Zusammenspiel mit einkriterieller Selektion einzuschätzen. Gleichzeitig können wir feststellen, dass die kombinierte Anwendung der Einflussfaktoren auf einem sehr einfachen Problem zu insgesamt guten Lösungen führt und damit einen vielversprechenden Weg für die modulare Kombination von lokalen Variationseinflüssen aufzeigt.

In diesem kurzen Kapitel werden wir in einer Vorstudie die Übertragbarkeit des Ansatzes auf verschiedene (aber noch recht einfache) Probleme evaluieren und daraus die Notwendigkeit einer Weiterentwicklung des Räuber-Beute-Modells ableiten. Dazu werden wir keinen umfangreichen und kompetitiven Vergleich mit anderen Algorithmen der evolutionären mehrkriteriellen Optimierung durchführen, sondern in verschiedenen Testfällen zuerst das Verhalten der Lösungsbildung des Modells untersuchen.

### 6.1 Testprobleme und Versuchsaufbau

Für eine erste Betrachtung der Fähigkeiten und Einschränkungen des Räuber-Beute-Modells betrachten wir initial drei verschiedene Probleme unterschiedlicher Schwierigkeit. Zwei der Probleme gehören der von Coello et al. (2007) zusammengetragenen MOP Testproblemsammlung an und sind in Tabelle 6.1 definiert. Das erste Problem stellt eine leicht modifizierte Variante des zuvor verwendeten Problems  $\mathcal{F}_{\text{circ}}^2$  dar. Die Besonderheit des ersten Testproblems von Binh & Korn (1996) besteht in der Orientierung der Paretomenge. Während die Paretomenge des Pro-

TABELLE 6.1: Definition der in diesem Kapitel verwendeten Testprobleme, um die Grenzen des Räuber-Beute-Modells zu verdeutlichen. Für die Analyse und Visualisierung der Effekte beschränken wir uns auf zweidimensionale Suchräume.

Name	Testproblem	Initialisierung
Binh & Korn (1996)	$f_1(x) = x_1^2 + x_2^2$ $f_2(x) = (x_1 - 5)^2 + (x_2 - 5)^2$	$\vec{x} \in [-5, 10]^2$
Fonseca & Fleming (1995)	$f_1(x) = 1 - \exp\left(-\sum_{i=1}^n (x_i - \frac{1}{\sqrt{(n)}})^2\right)$ $f_2(x) = 1 - \exp\left(-\sum_{i=1}^n (x_i + \frac{1}{\sqrt{(n)}})^2\right)$	$\vec{x} \in [-4, 4]^2$
Kursawe (1999)	$f_1(x) = 10 - g(x)$ $f_2(x) = 10 + \sum_{i=1}^n  x_i ^{0.8} + 5 \cdot \sin^3(x_i)$ $g(x) = \sum_{i=1}^{n-1} 10 \cdot \exp\left(-0.2 \cdot \sqrt{x_i^2 + x_{i+1}^2}\right)$	$\vec{x} \in [-5, 5]^2$

blems  $\mathcal{F}_{\odot}^2$  durch eine parallel zur Achse der Entscheidungsvariablen  $x_1$  verlaufende Gerade repräsentiert wird, verläuft die Paretomenge von Binh als Gerade zwischen den Punkten  $(0,0)^T$  und  $(5,5)^T$  im Suchraum. Die Paretofront ist eine konvexe Kompromisskurve analog zu jener von  $\mathcal{F}_{\odot}^2$ . Das zweite betrachtete Problem von Fonseca & Fleming (1995) wird durch eine konvexe aber zweidimensionale Menge im Suchraum ausgezeichnet und besitzt eine von der Anzahl der betrachteten Entscheidungsvariablen unabhängige konkave Paretofront. Das dritte Problem von Kursawe (1999) besteht aus einer nicht-konvexen und nicht zusammenhängenden Paretomenge. Die resultierende Paretofront weist konkave und konvexe Teilstücke auf.

Für die Durchführung der Evaluation werden für über alle Probleme weitgehend gleiche Versuchsanordnung gewählt, die sich untereinander nur in der Kombination der gewählten Räuber-Spezies unterscheiden. Grundsätzlich werden dem Räuber-Beute-Modell unabhängig von der Räuberanzahl insgesamt 1.800 Funktionsauswertungen erlaubt. Der Torus wird für eine bessere Abdeckung durch wenige Räuber und zur Beschleunigung der Simulation auf eine Größe von  $15 \times 15 = 225$  Individuen reduziert, die jeweils in dem in Tabelle 6.1 angegebenen Bereich für das betrachtete Testproblem zufällig initialisiert werden. Für das Testproblem von Binh werden für jedes Kriterium jeweils ein mit Mutation und drei mit Simplex-Rekombination ausgestattete Räuber eingesetzt. Für die übrigen beiden Probleme werden jeweils ein die Mutation tragender und zwei den Simplex-Operator tragende Räuber verwendet. Für die Mutation wird eine Schrittweite von  $\sigma = 0,05$  festgelegt. Jeder Räuber bewegt sich mit einer gleichverteilten Zufallsschrittweite von 1 und spannt eine Selektions- und Reproduktionsnachbarschaft mit Nachbarschaftsradius  $\nu = 1$  auf. Alle hier verwendeten Parameter, etwa die Anzahl der Räuberindividuen verschiedener Spezies für ein Problem, ebenso wie die Ein-

stellungen der festen Mutationsschrittweite sind auf Basis von Vorexperimenten festgelegt worden.<sup>1</sup>

Es sei noch darauf hingewiesen, dass sich die Bildung von Lösungen beim Räuber-Beute-Modell in verschiedenen Phasen abspielt, so dass zu unterschiedlichen Simulationszeitpunkten verschiedene Gebiete der Lösungsmenge stark ausgeprägt sind, sich dort also vermehrt Lösungen sammeln. Dies hängt insbesondere mit der zufälligen Bewegung der Räuber auf der Population und dem damit unterschiedlich stark wirkenden Einfluss derselben zusammen. Aus diesem Grunde wurde für die Konservierung guter Lösungen ein externes Archiv genutzt. Dieses wirkt explizit *nicht* in den Evolutionsprozess zurück, wie es etwa bei den in Abschnitt 3.2.1 beschriebenen Methoden üblich ist. Das hier eingesetzte Archiv ist vollständig unabhängig vom Räuber-Beute-Modell und ausschließlich zur Erhaltung bereits gefundener Lösungen vorgesehen. Typische Paretofronten der mehrfach durchgeführten Experimente (20 Experimente pro Problem) sind in Abbildung 6.1 dargestellt und werden im folgenden Abschnitt erläutert.

## 6.2 Ergebnisse

Die Resultate der Simulationen in Abbildung 6.1 zeigen direkt, dass das Räuber-Beute-Modell mit der modularen Kombination von Mutation und Rekombination für die Probleme von Binh und Fonseca eine gute Abdeckung und ebenfalls eine gute Annäherung an die Paretofront ermöglicht. In den Grafiken ist die jeweilige Paretofront grau eingezeichnet. Es ist also möglich zu  $\mathcal{F}_{\otimes}^2$  ähnliche Probleme mit dem Ansatz gut zu lösen. Betrachten wir hingegen die exemplarische Darstellung der Paretofront für das Kursaweproblem, so können wir zweierlei feststellen: Der untere Teil der Paretofront ist gut abgedeckt und relativ exakt approximiert, der obere Teil hingegen weist keine gute Approximationsqualität auf. Der entsprechende Teil der Paretofront ist in Abbildung 6.1 durch eine gestrichelte Box kenntlich gemacht. Diese Aufteilung deutet darauf hin, dass Teile der Paretomenge durch das Räuber-Beute-Modell im Gegensatz zu den beiden übrigen Problemen nicht abgedeckt werden können.

Eine genauere Betrachtung der Paretomenge der jeweiligen Probleme bestätigt diese Annahme. Abbildung 6.2 stellt dazu exemplarisch eine Momentaufnahme der Lösungen im Suchraum nach Abschluss der Simulationen dar. Wiederum sind die Paretomengen in Grau gekennzeichnet. Während die Paretomengen für die Probleme von Binh und Fonseca weitgehend vollständig von den approximierten Lösungen abgedeckt werden, ist die Lösungsmenge von Kursawes Problem nur teilweise approximiert worden. Der in der gestrichelten Box dargestellte Teil

---

<sup>1</sup>Dabei wurde durch wiederholte Experimente und durch die Abtastung verschiedener Parametereinstellungen empirisch eine „gute“ Einstellung bestimmt. Es wurden keine Parameteroptimierungsmethoden – wie etwa von Bartz-Beielstein, Lasarczyk & Preuss (2005) vorgeschlagen – angewendet, da die hier durchgeführten Experimente nicht zur Bestimmung der besten Parameterkonfiguration, sondern ausschließlich für die prinzipielle Bewertung des algorithmischen betrachtet werden.

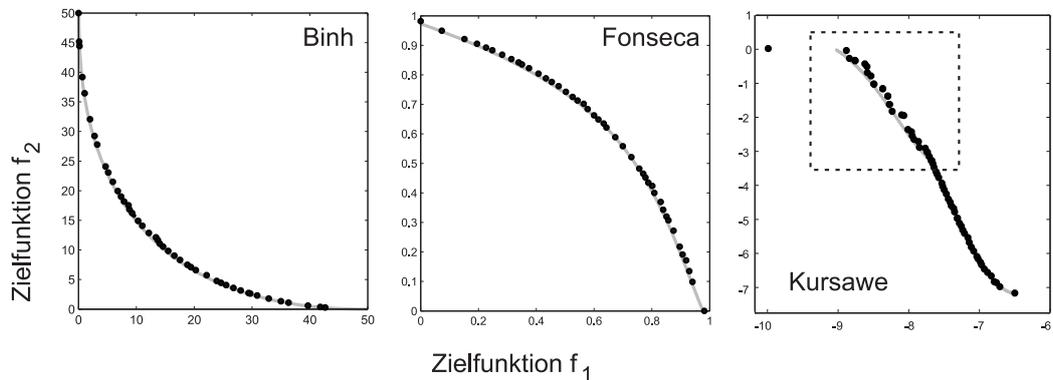


ABBILDUNG 6.1: Darstellung typischer Paretofronten bei Anwendung des Räuber-Beute-Modells auf die Probleme von Binh, Fonseca und Kursawe. Die tatsächliche Paretofront ist in der Grafik als graue Linie dargestellt. Während der Berechnung wurde ein externes Archiv zur Konservierung erreichter Lösungen eingesetzt.

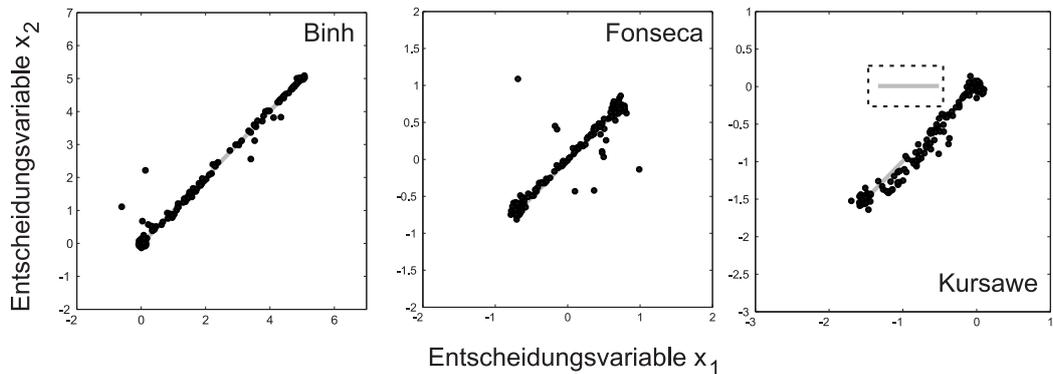


ABBILDUNG 6.2: Darstellung der zu den in Abbildung 6.1 ermittelten Paretomengen. Die Darstellung ist jeweils eine Momentaufnahme der Population am Ende der Simulation.

der Paretomenge ist überhaupt nicht abgedeckt. Dieser Teil ist aber insbesondere für die Abdeckung der oberen Region der Paretofront verantwortlich. Diejenigen Lösungen, die in Abbildung 6.1 (rechts) in diesem Bereich dargestellt sind, sind ausschließlich aufgrund der externen Archivierung in den ersten Schritten des Evolutionsprozesses noch vorhanden. In der endgültigen Paretomenge sind diese Lösungen verschwunden, also offenbar instabil innerhalb der Population.

### 6.3 Diskussion und kritische Betrachtung

Die Ergebnisse der Vorstudie zeigen, dass die Leistungsfähigkeit des Räuber-Beute-Modells sehr differenziert betrachtet werden muss. Während es bei einigen Problemstellungen gut anwendbar zu sein scheint, weist die Approximation bei anderen Problemen Schwächen auf. Wir wollen beide Aspekte kurz beleuchten.

### 6.3.1 Modulare Lösungsbildung im Suchraum

Die beispielhafte Untersuchung der Testprobleme  $\mathcal{F}_{\odot}^2$  in Abschnitt 5.2.4, Binh und Fonseca, hat gezeigt, dass die Kombination verschiedener Variationsoperatoren auf der Population durch Kopplung an die Räuber zu einer guten Approximation der Lösungsmenge führen kann. Voraussetzung ist jedoch immer, dass die Zielmenge einen konvexen Charakter aufweist. Bei allen drei Problemen ist die Pareto-menge durch eine Gerade zwischen den beiden Extremlösungen des bi-kriteriellen Problems gegeben. Hier können – entsprechend den Ergebnissen der Analyse in Kapitel 5 – Mutation und Rekombination ihren jeweiligen Einfluss zur Lösungsbildung beitragen: Während die Mutation in Kombination mit der einkriteriellen Selektion zu einer Ausbildung der Extremlösungen führt, tragen die mit Simplex-Rekombination ausgestatteten Räuber zu einer Lösungsbildung zwischen den Extremlösungen bei.

Interessant ist an dieser Stelle die Eigenschaft des Räuber-Beute-Modells, vor allem die Lösungsmenge eines Problems zu betrachten: In der Literatur beschränkt sich die Betrachtungsweise von Testproblemen (siehe etwa bei Coello et al. (2007)) in der Regel auf die Form der Paretofront. Dies ist im Kontext von Dominanz- und Indikator-basierten Verfahren verständlich, da diese Verfahren insbesondere durch die Form der Paretofront beeinflusst werden. Das Räuber-Beute-Modell hingegen betrachtet keinen globalen Zusammenhang der Lösungen in der Paretofront, sondern ausschließlich einzelne Kriterien, so dass die Form der Paretofront insgesamt keine Rolle spielt. Vielmehr wird von jedem einzelnen Räuber angestrebt, Lösungen im Suchraum gemäß seines Kriteriums anzuordnen. Um in diesen Vorgang einzugreifen und in der Gesamtheit des Modellverhaltens einen positiven Effekt zu erzielen, ist es wichtig, die an die Räuber gebundenen Einflussfaktoren gut zu wählen und zu kombinieren. Dies impliziert direkt die große Bedeutung von Variationsoperatoren. Grimme & Lepping (2007) haben in ihrer Arbeit dargestellt, dass die Entwicklung von Variationsoperatoren für die mehrkriterielle evolutionäre Optimierung nur geringe Beachtung erfährt. Als eines der wenigen Beispiele wird die Arbeit von Rudenko & Schoenauer (2004) erwähnt.<sup>2</sup> Auch die Simplex-Rekombination stellt durch ihre Fähigkeit zur Kontraktion der Lösungsmenge auf eine beliebig im Suchraum angeordnete konvexe Form eine wichtige Ergänzung im Kontext mehrkriterieller Variationsoperatoren über die intermediäre Rekombination hinaus dar. So ist es möglich, durch ihre Anwendung in Kombination mit fester Mutation im Räuber-Beute-Modell eine große Anzahl von Testproblemen zu lösen, solange diese eine konvexe Lösungsmenge im Suchraum aufweisen. Zu diesen Problemen gehören auch die weit verbreiteten Testprobleme von Zitzler, Deb & Thiele (2000) aus der sogenannten ZDT-Suite. Obwohl diese Probleme unterschiedliche

<sup>2</sup>Diese Autoren schlagen einen Dominanz-basierten Crossover-Operator vor, bei dem eine Paarungsrestriktion derart besteht, dass sich die besten Individuen nur mit echt dominierten Kandidaten vermischen dürfen. Dahinter verbirgt sich die Hoffnung, dass sich eine Tendenz in Richtung der Paretofront herausbildet und weitervererbt wird. Hier wird jedoch wieder die Paretofront als maßgeblicher Referenzpunkt angesehen. Die Partomenge und der Suchraum selbst spielen eine untergeordnete Rolle.

Paretofronten (konvex bis konkav) besitzen und mit zunehmender Dimensionalität des Suchraums mehr lokale Optima aufweisen und damit aufwändiger zu approximieren sind, besteht die Paretomenge bei jedem Problem aus einer Geraden mit  $x_1 \in [0.1, 1]$  und  $x_i = 0, \forall i : 2 \leq i \leq n$ . Die Definition der Testprobleme ist in Anhang A.2 angegeben. Einige Ergebnisse des Räuber-Beute-Modells für diese Probleme werden später in Kapitel 8 für Vergleichszwecke genutzt und zeigen, dass prinzipiell viele konvexe Mengen durch eine entsprechende Kombination von Mutation und Simplex-Rekombination angenähert werden können. Die Qualität hängt oftmals aber von der Dimensionalität, der Komplexität der einzelnen Kriterien (unimodal oder multimodal) und dem Aufwand an Zielfunktionsauswertungen ab.

### 6.3.2 Die Problematik nicht-konvexer Probleme

Für eine große Anzahl von Testproblemen in der Literatur gilt, dass bei ihrer Entwicklung oftmals unterschiedliche Komplexität in der Paretofront (konvex, konkav, viele lokale Fronten) im Mittelpunkt der Betrachtung stand. Eine Berücksichtigung von unterschiedlich komplexen Paretomengen ist meist nicht Gegenstand der Überlegungen. Dabei hat unser Experiment auf dem Problem von Kursawe gezeigt, dass insbesondere die Probleme mit nicht-konvexen Paretomengen für das Räuber-Beute-Modell eine große Hürde darstellen und nicht ausreichend approximiert werden können. Zugleich werden Lösungen stabilisiert, die nicht zur optimalen Lösungsmenge gehören, siehe Abbildung 6.2. Für eine genaue Bewertung des Modells und zur Ermittlung der Gründe für die beobachteten Effekte, betrachten wir schematisch die optimale Lösungsmenge des Kursawe-Problems und die Komponenten aus denen sich das Problem zusammensetzt. In Abbildung 6.3 sind die wichtigen Komponenten für unsere Betrachtung dargestellt. Die optimale Lösungsmenge des Kursawe-Problems (links) besteht aus einer einzelnen Lösung im Punkt  $(0, 0)$ , der in der Skizze mit  $b$  bezeichnet ist, und ein globales Optimum für  $f_1$  darstellt. Weiterhin umfasst die Paretomenge die grau gefärbte Menge zwischen  $c$  und  $d$ , sowie die schwarz markierte Menge startend bei  $a$ . Der Punkt  $a$  ist das globale Optimum für  $f_2$ . Die entsprechend eingefärbten Teile der Paretofront (rechts) zeigen an, welcher Teil der Lösungsmenge welchen Teil der Paretofront bildet. Zusammen mit den Abbildungen 6.1 und 6.2 ist offensichtlich, dass der Ansatz den oberen (grauen) Teil der Paretofront nicht erreicht. Stattdessen wird die gefundene Lösungsmenge auf der Verbindungslinie zwischen  $a$  und  $b$  ermittelt. Wie in Abbildung 6.3 (rechts) leicht zu erkennen ist, stellen die dadurch im Bildraum erzeugten Lösungen jedoch dominierte Lösungen dar (gestrichelt).

Das beobachtete Verhalten lässt sich direkt als eine Auswirkung der Kombination von Mutation und Simplex-Rekombination, wie sie in Abschnitt 5.2 untersucht wurden, erklären. Zusätzlich zu der durch die einkriterielle Selektion induzierten Bevorzugung von Extremlösungen innerhalb der Population, trägt die Mutation entsprechend der Folgerungen 5.3 und 5.2 zu einer Stabilisierung der Extremlösungen bei. Dies geschieht auch für Kursawes Testproblem. Da die Punkte  $a$  und  $b$  im

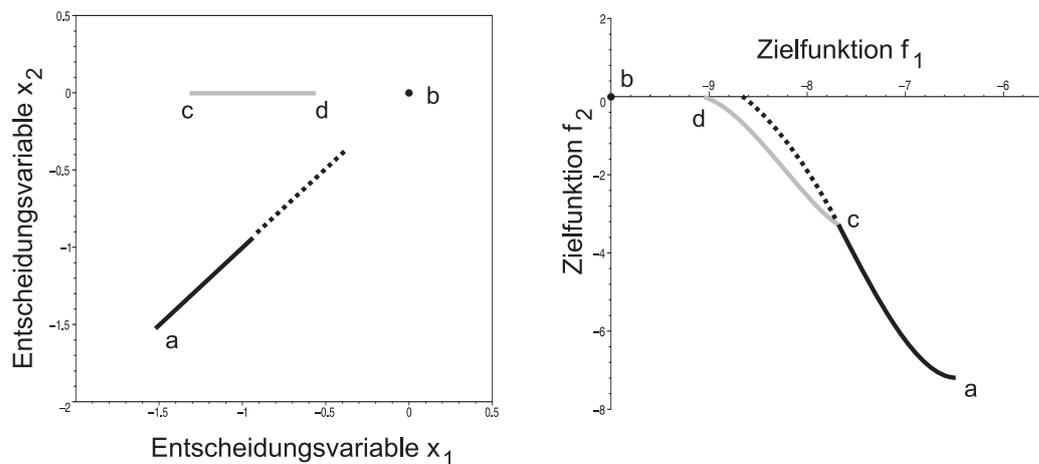


ABBILDUNG 6.3: Skizze zur Zusammensetzung des Kursawe-Problems in Paretomenge (links) und Paretofront (rechts). Die Teilabschnitte der Paretofront sind analog zu den entsprechenden Lösungen im Suchraum in unterschiedlichen Stricharten dargestellt. Die gestrichelte Linie gehört dabei nicht zur Paretomenge, stellt aber einen Teil der vom Räuber-Beute-Modell ermittelten Lösungsmenge dar.

Suchraum die globalen Optima der beiden Kriterien darstellen, werden diese durch die Mutation bevorzugt und stark ausgebildet. Die Rekombination trägt nun entsprechend Folgerung 5.5 zur Mittlung zwischen diesen beiden Lösungen bei. Dabei entstehen ausschließlich Lösungen, die innerhalb der konvexen Hülle der beiden Extrempunkte, also auf der direkten Verbindungslinie, liegen. Die Punkte  $c$  und  $d$  hingegen, stellen lokale Optima der jeweiligen Ziele des Testproblems dar und werden gemäß Folgerung 5.2 durch die Bildung stabiler Gebiete bezüglich der globalen Optima aus der Population verdrängt.

Aufgrund dieser Beobachtung können wir feststellen, dass das Räuber-Beute-Modell in seiner vorliegenden Form sicherlich für die Behandlung konvexer und damit relativ gutmütiger Probleme geeignet ist. Eine Anwendung auf Probleme mit nicht-konvexer Paretomenge garantiert jedoch keine hinreichende Lösungsqualität. Vielmehr strebt die Population – wie bereits oben beschrieben – durch den Einfluss der einkriteriellen Selektion und verstärkt durch den analysierten Effekt der Mutation hin zu Extrem Lösungen. Die konvexe Operation der Rekombination kann dies zwar kompensieren, jedoch nur für Probleme, deren Lösungsmenge durch eine konvexe Menge beschrieben wird.

Wir können also abschließend feststellen, dass der vorgeschlagene modulare Ansatz zwar Flexibilität zum Modell hinzufügt und auch die Integration von verschiedenen lokalen Operatoren erlaubt, jedoch keinen Lösungsansatz für die Kompensation der starken einkriteriellen Selektion bietet. Für eine Nutzung des Räuber-Beute-Modells in einem größeren Kontext von Problemen müssen daher Möglichkeiten gefunden werden, andere Komponenten des Modells einzusetzen oder anzupassen. Dabei ist es wichtig die Komplexität des Modells nicht grundsätzlich zu erhöhen und den modularen, dezentralen und gut parallelisierbaren Aufbau bei-

zubehalten. In den nächsten beiden Kapiteln betrachten wir sowohl die Beute-  
wie auch die Räuberpopulation als mögliche Kandidaten für eine Erweiterung des  
Ansatzes.

## Kapitel 7

# Integration Gradienten-basierter Suche

Die Diskussion im vorherigen Kapitel hat gezeigt, dass insbesondere die Stabilisierung von Teilen der nicht-konvexen Paretomengen einiger Testprobleme eine Herausforderung für das Räuber-Beute-Modell darstellt. Der globale Mechanismus der einkriteriellen Selektion toleriert keine lokal optimalen Lösungen, auch wenn diese zur Paretomenge gehören. An dieser Stelle ist es denkbar, den Einfluss der globalen Selektion durch lokale Einflüsse zu reduzieren und damit zum Erhalt der gewünschten Lösungen beizutragen. Bereits die Betrachtung der Grundlagen der ein- und mehrkriteriellen Optimierung in Kapitel 1 hat gezeigt, dass die Verfolgung der (negativen) Gradientenrichtung – also der Richtung des steilsten Abstiegs – ein vielversprechender Weg zum schnellen Erreichen einer Pareto-optimalen Lösung sein kann.

Die Einbeziehung von Gradienteninformationen in die iterative Suche ist ein Vorgehen, das bereits lange vor der Entwicklung evolutionärer mehrkriterieller Verfahren oder der algorithmischen Untersuchung mehrkriterieller Problemstellungen eingesetzt wurde. In diesem Zusammenhang sei auf die ausführliche Darstellung solcher Verfahren bei Schwefel (1995) verwiesen.<sup>1</sup> Auch im Kontext mehrkriterieller Optimierung werden Gradienteninformationen direkt oder approximativ eingesetzt, um einen Algorithmus bei der Annäherung an die Paretofront zu unterstützen und auch zu leiten. Dabei wird oftmals Fritz John's notwendige Bedingung für Pareto-Optimalität aus Satz 1.10 angewendet, um einen Indikator für die Annäherung an die (lokale) Paretofront zu entwickeln. Eine frühe Anwendung dieser Eigenschaft ist bereits bei Ester (1987) zu finden. Er entwickelte eine iterative Strategie, die auf der in Beispiel 1.11 angewendeten Technik paralleler Gradienten basiert, jedoch global optimale Lösungen nur auf konvexen Problemen findet. Im Bereich der mehrkriteriellen evolutionären Optimierung legen Brown & Smith (2005) die theoretischen Grundlagen für die Integration von Gradienten-

---

<sup>1</sup>Von Schwefel werden etwa Steepest Descent, PARTAN oder Powells konjugierte Richtungen betrachtet.

teninformationen und definieren die auch in dieser Arbeit verwendeten mehrkriteriellen Kegel für Abstieg, Aufstieg und widersprechende Richtungen (descent, ascent und contradictory cones). Vorbereitend für den späteren Einsatz innerhalb globaler Strategien entwickeln Harada, Sakuma & Kobayashi (2006) eine lokale Suchstrategie, die unter Annahme von Differenzierbarkeit der Kriterien den steilsten Abstieg verfolgt. Sie erweitern sie um die Fähigkeit der Grenzenbehandlung (constraint handling) und evaluieren sie im Vergleich zu anderen Methoden. Dahingegen konstruieren Bosman & de Jong (2006) ein Verfahren, mit dem verschiedene Gradientenabstiegsverfahren adaptiv durch einen globalen mehrkriteriellen Optimierungsalgorithmus, in Abhängigkeit vom erwarteten Fortschritt, aktiviert oder deaktiviert werden. Solche Ansätze werden oftmals als Hybridisierung bezeichnet. Shukla (2007) evaluiert ebenfalls mehrere den Gradientenabstieg approximierende Verfahren und integriert diese als Mutationsoperatoren oder lokale Suchstrategien in NSGA-2. Neben einigen vom Autor festgestellten Verbesserungen im Ergebnis der Approximation zeigt er auch, wie Fritz John's Regel als Abbruchkriterium für den Algorithmus selbst verwendet werden kann. Schließlich verwenden auch Schuetze, Sanchez & Coello Coello (2008) die grundsätzliche Idee des Gradientenabstiegs, wobei sie nicht explizit Gradienten approximieren, sondern die im Algorithmus verfolgte Abstiegsrichtung aus der gegenseitigen Dominanz von Individuen bestimmen. Im Falle keiner klaren Dominanzbeziehung zweier Lösungen, wird dieser Mechanismus zum sogenannten „hill climber with side steps“, trägt also zusätzlich zur Annäherung an die Paretofront zur Diversitätserhaltung bei. Die Autoren integrieren ihren Ansatz in SPEA2.

Für die Integration von Gradienteninformationen in das Räuber-Beute-Modell entwickeln wir ein einfaches Verfahren, das von einigen Aspekten der zuvor diskutierten Arbeiten inspiriert ist. Nach einer kurzen Erläuterung zum Verfahren der Gradientenapproximation entwerfen wir eine einfache Kombinationstechnik für die einkriteriellen Gradienten und diskutieren den Ansatzpunkt für die Integration in den Räuber-Beute-Ansatz. Abschließend evaluieren wir den Ansatz experimentell. Insgesamt stützen wir uns bei dieser Betrachtung verstärkt auf die Arbeit und Ergebnisse von Grimme, Lepping & Papaspyrou (2009).

## 7.1 Approximation der Gradienteninformation

Ist eine Funktion differenzierbar, so kann durch Bildung der partiellen Ableitung der Gradient eines Vektors im Suchraum exakt bestimmt werden. Oftmals ist diese Methode jedoch im Kontext praktischer evolutionärer Verfahren nicht anwendbar, da die betrachteten Zielfunktionen entweder nicht differenzierbar oder sogar unbekannt sind. Es muss also auf eine Approximation der Gradientenrichtung zurückgegriffen werden. Für die numerische Approximation eines Gradienten wird gewöhnlicherweise der Differenzenquotient in jeder Komponente berechnet und damit eine relativ genaue Abschätzung erreicht. Dies erfordert jedoch bei  $n$  Dimensionen  $2n$  Funktionsauswertungen, was im Falle von  $m$  Kriterien zu  $\mathcal{O}(m \cdot n)$  Auswertungen

für die Bestimmung einer Abstiegsrichtung führt. Um die Anzahl der Auswertungen möglichst gering zu halten, vereinfachen wir die von Spall (1992, 1998) vorgeschlagene und implementierte Methode der gleichzeitigen Störung (engl.: simultaneous perturbation method). Spall schlägt für die Approximation eines Gradienten  $\tilde{g}_i$  die beidseitige Störung eines Vektors  $\vec{x}$  durch einen zufällig ermittelten und durch ein Skalar  $\sigma$  skalierten normalisierten Störungsterm  $\vec{e}$  vor, siehe Gleichung 7.1. In Spalls Arbeit wird  $\vec{e}$  als Bernoulli $\pm 1$ -verteilt angenommen.

$$\tilde{g}(\vec{x}) = \frac{f(\vec{x} + \sigma\vec{e}) - f(\vec{x} - \sigma\vec{e})}{2\sigma} \quad (7.1)$$

Wir vereinfachen dieses Vorgehen ausgehend von Vektor  $\vec{x}$  durch die annähernd gleichverteilte Wahl von  $t$  Punkten auf der Oberfläche einer  $n$ -dimensionalen Hyperkugel mit Radius  $r \in \mathbb{R}$  (gewöhnlich  $0 < r \ll 1$ ) um  $\vec{x}$ , siehe Abbildung 7.1. Ein

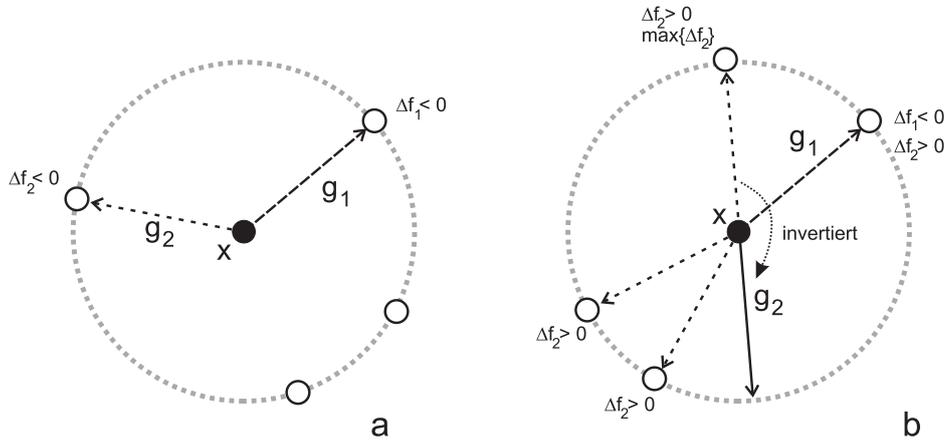


ABBILDUNG 7.1: Skizze des verwendeten Mechanismus zur Approximation der Gradientenrichtungen für zwei Kriterien mit  $t = 4$  Explorationspunkten. Auf einer Hyperkugel (hier Kreis) um  $\vec{x}$  werden zufällig 4 Punkte generiert und pro Kriterium die Richtung des steilsten Abstiegs als Gradient gewählt,  $g_1$  als Gradient für Kriterium 1 und  $g_2$  für Kriterium 2. Wenn keine Abstiegsrichtung gefunden wird, so wird die Gegenrichtung des stärksten Anstiegs gewählt, siehe b.

zufälliger Vektor  $\vec{q} = (q_1, \dots, q_i, \dots, q_n)$  auf der Oberfläche der Kugel ergibt sich über die Vorschrift für  $n$ -dimensionale Kugelkoordinaten in Gleichung 7.2, wobei die Winkel  $\lambda_2$  und  $\lambda_k, k = 3, \dots, n$  aus unterschiedlichen Bereichen gleichverteilt gewählt werden.

$$q_i = x_i + r \cdot \cos(\lambda_i) \cdot \prod_{j=i+1}^n \sin \lambda_j \quad (7.2)$$

mit  $i = 1, \dots, n, \lambda_1 = \pi, \lambda_2 = \mathcal{U}(0, 2\pi)$

und  $\lambda_k = \mathcal{U}(-\pi, \pi), k = 3, \dots, n$

Um zu bestimmen, ob es sich bei der gewählten Richtung um eine Abstiegsrichtung handelt, wird ausschließlich überprüft, ob der Funktionswert  $f(\vec{q})$  geringer

als  $f(\vec{x})$  ist. Die Differenz der Werte wird als  $\Delta f = f(\vec{q}) - f(\vec{x})$  dargestellt. Gilt  $\Delta f < 0$ , so handelt es sich um eine Abstiegsrichtung für  $f$ , andernfalls um eine Aufstiegsrichtung.

Die genutzte Methode wird in zweierlei Hinsicht parametrisiert. Einerseits ist es notwendig, den Radius  $r$  der Hyperkugel festzulegen. Andererseits können für die Bestimmung der endgültigen Gradientenapproximation  $t$  verschiedene Explorationen um  $\vec{x}$  vorgenommen werden. Im Allgemeinen wird aus allen Approximationen dann die Richtung des größten Abstiegs gewählt. In Abbildung 7.1(a) sind die entsprechend ermittelten Gradienten  $g_1$  und  $g_2$  für die Zielfunktionen  $f_1$  und  $f_2$  beispielhaft dargestellt. Die zugehörigen Explorationspunkte bieten für die jeweilige Zielfunktion den größten Abstieg.

Natürlich kann bei der zufälligen Ermittlung der Explorationspunkte die Situation auftreten, dass nur Aufstiegsrichtungen bestimmt wurden. Abbildung 7.1(b) zeigt diesen Fall beispielhaft für eine Zielfunktion  $f_2$ . In diesen Fall wird die Gegenrichtung des maximalen Aufstiegs gewählt. Bei geringem  $r$  gehen wir davon aus, dass diese Richtung einen Abstieg bezeichnet.<sup>2</sup> Abweichend von der Methode von Spall verlassen wir uns hier nur auf eine einseitige Störung zur Bestimmung des Gradienten. Dies kann in der numerischen Auswertung zu einer Auslöschung<sup>3</sup> führen und die Gradienteninformation stark verfälschen. Zugleich spart diese Variante aber eine zweite Auswertung der Zielfunktion, da die Bewertung des betrachteten Vektors  $\vec{x}$  grundsätzlich im Zuge des evolutionären Prozesses erfolgt. Ähnlich wie Spall verlassen wir uns jedoch darauf, dass die Gradientenrichtung durch wiederholte Anwendung des Verfahrens über den Suchverlauf angenähert wird. Aus diesem Grund erlauben wir, wie Spall, ebenfalls eine Wahl von  $t < n$ . Damit reduzieren wir – im Gegensatz zur Bildung des Differenzenquotienten – die Anzahl der benötigten Funktionsauswertungen für die Bestimmung der Gradienteninformation deutlich.

## 7.2 Bestimmung des mehrkriteriellen Abstiegs

Mit dem Satz 1.10 von Fritz John und dem approximierten Gradienten für jedes Kriterium sind wir nun in der Lage, für ein gegebenes mehrkriterielles Problem ein lokales Abstiegsgebiet zu definieren. Darüber hinaus können wir noch zwei andere Gebiete im Suchraum definieren:

**Abstieg:** Das durch den sogenannten Abstiegskegel (engl.: descent cone) definierte Gebiet beschreibt jene Richtungen im Suchraum, die einen Fortschritt, also

<sup>2</sup>Natürlich ist diese Festlegung eine optimistische Abschätzung. Sie erspart uns jedoch zusätzliche Auswertungen.

<sup>3</sup>Die Auslöschung bezeichnet in der Numerik den bei der Subtraktion zweier Gleitkommazahlen auftretenden Effekt des Genauigkeitsverlustes. Werden zwei Zahlen voneinander subtrahiert, die sich erst in späteren Nachkommastellen unterscheiden, kann die verbleibende Differenz das ausschließliche Resultat vorangegangener Rundungsfehler sein.

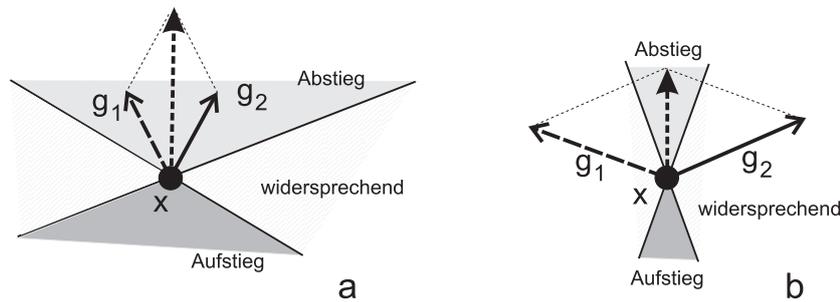


ABBILDUNG 7.2: Schematische Darstellung des verwendeten lokalen Suchverfahrens. Auf Basis der ermittelten Gradienten werden Gebiete des gemeinsamen Abstiegs, der widersprechenden Richtung und des Aufstiegs bestimmt. Wenn die Gradienten annähernd in die gleiche Richtung zeigen (siehe a), so ist das Gebiet des gemeinsamen Abstiegs groß und  $\vec{x}$  weit von der lokalen Paretomenge entfernt. Sind die Gradienten annähernd antiparallel (siehe b), so befindet sich  $\vec{x}$  nah an der lokal optimalen Lösung und kooperativer Abstieg ist schwer.

eine Verringerung des Funktionswertes für jedes verfolgte Kriterium versprechen. Diese Richtungen sind natürlich vorzuziehen, um die lokale Paretomenge zu erreichen.

**Aufstieg:** Der sogenannte Aufstiegskegel (engl.: ascent cone) umfasst all jene Richtungen die zu einer Verschlechterung des Funktionswertes jedes Kriteriums führen. Sie führen von der lokalen Paretomenge weg und sollten nicht verfolgt werden.

**Widerspruch:** In dem sogenannten kontradiktorischen Kegel (engl.: contradictory cone) liegen all jene Richtungen, die zur Verbesserung der Funktionswerte einiger Kriterien beitragen, andere aber zugleich verschlechtern. Eine Bewegung entlang der jeweiligen Richtungen entspricht also grob einer Bewegung entlang der lokal Pareto-optimalen Front.

Abbildung 7.2 stellt diese Richtungskegel exemplarisch dar. Sie ergeben sich direkt über die Gradienteninformationen entsprechend Satz 1.10 aus den Orthogonalen der Abstiegsrichtung. Analog zu den Überlegungen von Ester (1987) können wir feststellen, dass der Abstiegskegel groß ist, wenn der Winkel zwischen den Abstiegsrichtungen für alle Kriterien gering ist. Dann zeigen alle negativen Gradienten in eine annähernd gleiche Richtung. Die Entfernung von der optimalen Lösungsmenge ist in diesem Falle relativ groß. Zeigen die Abstiegsrichtungen annähernd in entgegengesetzte Richtungen (antiparallel), so können wir von einer großen Nähe zur lokal optimalen Lösung ausgehen. Der Abstiegskegel wird in dieser Situation klein.

$$\vec{x}' = \vec{x} + \vec{z}, \text{ wobei } \vec{z} = \frac{1}{\sqrt{m}} \sum_{i=1}^m \omega_i \cdot \tilde{g}_i \quad (7.3)$$

$$\text{mit } \vec{z}, \tilde{g}_i \in \mathbb{R}^n \text{ und } \omega_i = \mathcal{U}(1, m), \sum_{i=1}^m \omega_i = m$$

Wir verwenden diesen Zusammenhang in Vorschrift 7.3, um den Schritt der lokalen Suche von  $\vec{x}$  nach  $\vec{x}'$  durchzuführen. Dort wird der Suchschritt  $\vec{z}$  durch eine gewichtete Kombination der normalisierten negativen Gradienten bestimmt. Nachdem die Gradienten kombiniert wurden, wird der resultierende Schritt entsprechend der Anzahl der Kriterien normalisiert, um große Schritte für die Situation  $m > 2$  zu verhindern. Zusätzlich wird jede Abstiegsrichtung mit einem zufälligen Gewicht bewertet, um den jeweiligen Einfluss der Richtung zu verstärken oder zu schwächen.

Ist  $\vec{x}$  weit von der optimalen Lösung entfernt, wird die resultierende Richtung  $\vec{z}$  den Abstiegskegel selten verlassen und somit zur Annäherung an die optimale Lösung beitragen. Liegt  $\vec{x}$  dicht an der Paretomenge, und ist der Abstiegskegel klein, so wird  $\vec{z}$  oftmals im kontradiktorischen Kegel liegen. Dies trägt zwar nicht zur Annäherung an die Paretomenge, wohl aber zur Diversitätserhaltung in der Nähe der optimalen Lösungsmenge bei. Zugleich ist es insbesondere dieser diversitätserhaltende Mechanismus, der konzeptionell dem durch die Räuber induzierten Extremalverhalten entgegengesetzt werden kann. Er sorgt – bildlich gesprochen – für eine Gegenbewegung zu der einkriteriellen Spezialisierung der Beuteindividuen, indem eine kontradiktorische Bewegung (entlang der Paretofront) ausgeführt wird.

Wir erwarten, dass durch die entsprechende Wahl der Suchrichtung ein Umschalten zwischen der Annäherung an die optimale Lösungsmenge und der Diversitätserhaltung in der Nähe der optimalen Lösungsmenge automatisch stattfindet. So passt sich der lokale Ansatz dann aufgrund der Gradienteninformationen automatisch an die Situation im Suchraum an.

### 7.3 Hybridisierung des Modells über die Beute

Wir folgen nun dem Grundgedanken, dass das Räuber-Beute-Modell aus verschiedenen Bausteinen besteht, und integrieren die lokale gradientenbasierte Suche in die Beuteindividuen. In der bisherigen Betrachtung waren die Räuber die einzigen agierenden Elemente des Ansatzes. Die Beute selbst war ausschließlich eine Entität zur Repräsentation von Lösungen. Nun erhält jede Beute die Fähigkeit, unabhängig von der durch die Räuber induzierten globalen Suche, für den durch sie repräsentierten Punkt im Suchraum eine Verbesserung entlang der Abstiegsrichtung anzustreben oder einer kontradiktorischen Richtung zu folgen. Dazu werden

für die entsprechende Lösung und für jedes Kriterium  $t$  verschiedene Explorationspunkte auf der Hyperkugel mit Radius  $r$  um die aktuelle Lösung bestimmt sowie anschließend gemäß Vorschrift 7.3 die Bewegungsrichtung festgelegt und ein Schritt dorthin ausgeführt. Eine Erfolgsüberprüfung im Sinne der Selektion findet hier nicht statt. Der gesamte lokale Suchprozess wird nebenläufig zu der Räuberaktion und der Aktivität anderer Beuteindividuen für jedes Beuteindividuum kontinuierlich wiederholt.

Es bleibt an dieser Stelle nur noch darzustellen, welche Auswirkungen die Integration innerhalb der Beuteindividuen auf das Gesamtmodell und dessen Modularität, Parallelisierbarkeit und Dynamik hat.

**Modularität und Parallelisierbarkeit:** Die Integration der lokalen Suche hat keine Auswirkungen auf den modularen Aufbau des Ansatzes. Alle Komponenten und deren Zusammenspiel bleiben in der bisherigen Form erhalten. Die zusätzliche Aktivität der Beute hat zwar lokale Effekte auf die Lösungsbildung, verändert jedoch nicht den grundlegenden Ablauf aus Algorithmus 1 (siehe Kapitel 4, Seite 61). Insbesondere kann die lokale Suche unabhängig von der algorithmischen Logik durchgeführt werden und im Sinne der Parallelisierung (vgl. Abschnitt 5.1), auf der Ebene der problemunabhängigen Parallelisierung, nebenläufig zum globalen Suchprozess durchgeführt werden. Der Einfluss der lokalen Suche ist durch diese lose Kopplung vorwiegend indirekt und ist mit einer autonomen Weiterentwicklung von Lösungen, die gerade nicht von einem Räuber betrachtet werden, vergleichbar. Wie in einem biologischen System gibt es keine aktive Beeinflussung der Räuber durch die Beute. Die Beute ist nun jedoch in der Lage, sich ggf. selbst gegen Räubereinfüsse zu „immunisieren“. Dies bedeutet, dass die Beute dem Räubereinfluss auf lokaler Ebene (durch die nah an der lokalen Paretofront durchgeführte Seitwärtsbewegung) einen Diversität erzeugenden Einfluss entgegensetzt und somit zur Stabilisierung lokaler Lösungen beiträgt.

**Auswirkung auf die Dynamik:** Betrachten wir die Dynamik des Ansatzes unter der Annahme der Integration des Gradientenansatzes, so erwarten wir durch die lokalen Auswirkungen einen stabilisierenden Effekt. Der Suchmechanismus ist ohne Einwirkung der Räuber in der Lage, eine Beute weiter zu entwickeln und die Population insgesamt in Richtung einer zumindest lokal optimalen Front zu treiben. Dabei ist allerdings klar, dass der Räubereinfluss begrenzt werden muss. Konfiguriert man das Modell mit einer großen Anzahl von Räubern, so können die lokalen Suchfortschritte erneut von den Räubern überlagert und weitgehend unwirksam gemacht werden. Insgesamt teilt sich die Dynamik in lokale und globale Aufgaben der verschiedenen Akteure auf. Die gradientenbasierte Suche ist dabei für eine feingranulare Einstellung der Lösungsmenge zuständig. Sie profitiert sowohl von der die Konvergenz unterstützenden Abstiegsbewegung wie auch von der kontradiktorischen Bewegung zur Diversitätserhöhung und -erhaltung. Die global

agierenden Räuber tragen insbesondere durch die Auswirkungen der Mutation zum Auffinden global optimaler Gebiete bei. Diese Gebiete kann die lokale Suche bei falscher Initialisierung nicht eigenständig auffinden. Beiden Elementen kommt also eine wichtige Aufgabe zu, die in der Konfiguration des Algorithmus ausbalanciert werden muss.

Wir können insgesamt feststellen, dass sich die lokale Suche durch unser Vorgehen vollständig in das modulare Konzept des Räuber-Beute-Ansatzes einfügt und die Parallelisierbarkeit nicht beeinträchtigt. Vielmehr nutzt die lokale Suche die Nebenläufigkeit, um zusätzliche Informationen in den Evolutionsprozess einfließen zu lassen. Der nächste Abschnitt analysiert nun den Mechanismus detailliert und untersucht am Beispiel des Kursawe-Problems, inwieweit die Erweiterung zur Lösung nicht-konvexer Problemstellungen beitragen kann.

## 7.4 Evaluation des Hybridansatzes

Um die neu zum Räuber-Beute-Modell hinzugefügte lokale Suche genauer zu untersuchen, unterteilen wir die Evaluation in zwei Abschnitte. Zuerst werden wir ausschließlich die Funktionsweise der gradientenbasierten Suche und ihr Verhalten bei der Annäherung an die Lösungsmenge sowie in der Nähe der Paretomenge untersuchen. Im zweiten Teil betrachten wir exemplarisch das Kursawe-Problem aus Kapitel 6 und überprüfen den Beitrag der lokalen Suche zur Stabilisierung und Verbesserung der Lösung. Wir wählen dieses Problem, da es zwei wichtige Eigenschaften besitzt. Einerseits verfügt es über eine nicht zusammenhängende Paretomenge, und andererseits finden sich verschiedene lokal optimale Lösungsmengen im Suchraum, die zur vorzeitigen Konvergenz führen können.

### 7.4.1 Ausschließliche Betrachtung der lokalen Suche

Für die Analyse der Funktionsweise des entwickelten gradientenbasierten Suchverfahrens setzen wir erneut das relativ einfache Testproblem  $\mathcal{F}_{\odot}^2$  ein, da dieses Problem eine konvexe Lösungsmenge hat und wir erwarten können, dass unser Verfahren in der Lage ist, die optimale Lösungsmenge zu erreichen.

Für die Untersuchung beschränken wir uns auf die Betrachtung einer einzelnen Lösung und evaluieren deren Entwicklung über die Zeit. Dafür setzen wir fest, dass pro Evaluation 2000 Iterationen der lokalen Suche entsprechend Vorschrift 7.3 durchgeführt werden. Wir starten die lokale Suche immer von der Position  $(3, 3)$  im Suchraum und deaktivieren jegliche Räubereinflüsse. Neben der kumulativen visuellen Auswertung<sup>4</sup> der Ergebnisse definieren wir im Folgenden drei einfache Maße.

---

<sup>4</sup>Für die grafische Darstellung werden alle besuchten Zwischenlösungen dargestellt, was zu einer Visualisierung des Suchpfades führt.

**Länge der initialen Phase** Startet die Suche entfernt von der Paretomenge, so wird erwartet, dass die auf den Gradienteninformationen basierende Suche Schritte in kooperative Richtung, also innerhalb des Abstiegskegels, durchführt. Erst in der Nähe der (lokalen) Lösungsmenge wird dieser Abstieg beendet, und die Suche findet vorwiegend in kontradiktorischer Richtung statt. Um die Dauer der Annäherung an die Lösungsmenge abhängig von der Anzahl der Explorationen  $t$  und dem Explorationsradius  $r$  zu untersuchen, zählen wir die Schritte, die bis zur Annäherung an die Paretomenge benötigt werden. Dazu setzen wir fest, dass die Annäherung abgeschlossen ist, wenn es ein  $\hat{x} \in \mathcal{X}^*$  für die aktuelle Lösung  $\vec{x}$  gibt, so dass für die euklidische Distanz zur Lösungsmenge gilt:  $\|\hat{x} - \vec{x}\| \leq 10 \cdot r$ .

**Abdeckung in der  $x_1$ -Komponente** Zur Beurteilung der „Seitwärtsschritte“ in kontradiktorische Richtungen, also entlang der Paretomenge und Paretofront, betrachten wir die kumulative Abdeckung der  $x_1$ -Komponente für unser spezielles Problem  $\mathcal{F}_{\odot}^2$ . Dazu archivieren wir jede Zwischenlösung des lokalen Suchprozesses exklusive jener Schritte, die für die Annäherung an die Paretomenge benötigt wurden. Die Abdeckung ergibt sich dann aus der  $x_1$ -Komponente der konvexen Hülle aller archivierten Lösungen.

**Minimale Distanz zur Paretomenge** Für die Beurteilung des Konvergenzverhaltens der lokalen Suche betrachten wir, welche Annäherung an die optimale Lösungsmenge erreicht werden kann. Dazu wird die minimale Distanz über die archivierten Ergebnisse aller Suchschritte bestimmt. Zusätzlich können wir dieses Maß abwandeln, so dass wir über alle archivierten Lösungen nahe der Lösungsmenge die mittlere Ausdehnung in der  $x_2$ -Komponente bestimmen. Dies gibt uns weniger Aufschluss über die absolut erreichbare Annäherung als vielmehr einen Eindruck von der Lösungsdichte nahe der optimalen Menge.

Für die Auswertung betrachten wir verschiedene Einstellungen für die Parameter  $t$  und  $r$  der lokalen Suche. Insbesondere ist interessant, ob für die Gradientenapproximation eine geringe Anzahl an Explorationsversuchen ausreicht. Von verschiedenen Einstellungen des Explorationsradius erwarten wir eine Veränderung in der Konvergenzgeschwindigkeit und ggf. einen Einfluss auf die maximale Güte der Annäherung. Für  $t$  wählen wir als einfachsten Fall eine einzige Exploration sowie 10 Explorationen für eine sehr genaue Abschätzung des Gradienten. Den Explorationsradius wählen wir mit  $r \in \{0.01, 0.05, 0.1\}$ .

Tabelle 7.1 fasst die Ergebnisse für die verschiedenen Kombinationen der Parameter und für alle betrachteten Bewertungsmaße zusammen. Alle Ergebnisse sind Mittelwerte aus 30 unabhängigen Experimenten inklusive der zugehörigen Standardabweichung. Für alle Vergleiche wurden Signifikanzen auf Basis des Wilcoxon-Rangsummentests ( $p \leq 0.05$ ) berechnet, siehe auch in der Zusammenfassung von Dalgaard (2002).

Für den Explorationsradius können wir aus den vorliegenden Ergebnissen di-

TABELLE 7.1: Ergebnisse für die Einzelbetrachtung der gradientenbasierten Abstiegsmethode ohne Einfluss der Räuber. Die Ergebnisse beziehen sich auf 30 Wiederholungen eines Experiments für das Problem  $\mathcal{F}_{\odot}^2$  mit nur einem Individuum und kumulativer Betrachtung der Suchpunkte. Gegenstand der Überprüfung sind verschiedene Parametereinstellungen für den Hyperkugeleradius und für die Anzahl von Explorationspunkten. Ausgewertet wurde die Anzahl der Iterationen rein kooperativer Bewegungen hin zur Lösungsmenge (Initial Phase), die Ausdehnung der kumulativen Suchpunkte (Abdeckung in  $x_1$ ) und die minimale Distanz zur Paretomenge. Alle Werte sind Mittelwerte über 10 Versuche inkl. Standardabweichung.

Parameter	Initiale Phase	Abdeckung in $x_1$	min. Distanz zu $\mathcal{X}^*$
$r = 0.01,$ $t = 1$	$487.47 \pm 35.86$	$0.352 \pm 0.095$	$3.76 \cdot 10^{-5} \pm 3.16 \cdot 10^{-5}$
$r = 0.01,$ $t = 10$	$339.33 \pm 15.41$	$0.487 \pm 0.127$	$1.63 \cdot 10^{-5} \pm 1.59 \cdot 10^{-5}$
$r = 0.05,$ $t = 1$	$55.43 \pm 5.88$	$1.750 \pm 0.325$	$6.66 \cdot 10^{-5} \pm 5.55 \cdot 10^{-5}$
$r = 0.05,$ $t = 10$	$38.20 \pm 1.42$	$1.614 \pm 0.350$	$3.14 \cdot 10^{-5} \pm 3.24 \cdot 10^{-5}$
$r = 0.1,$ $t = 1$	$28.87 \pm 3.22$	$2.139 \pm 0.143$	$8.80 \cdot 10^{-5} \pm 1.33 \cdot 10^{-4}$
$r = 0.1,$ $t = 10$	$19.53 \pm 1.07$	$2.134 \pm 0.115$	$3.53 \cdot 10^{-5} \pm 3.80 \cdot 10^{-5}$

rekt ableiten, dass ein kleiner Radius offensichtlich mit einer geringeren Konvergenzgeschwindigkeit zur Lösungsmenge führt. Insgesamt führt die Wahl einer kleinen Explorationsschrittweite nicht zu einer signifikant besseren maximalen Annäherung an die Paretomenge, jedoch ist die durchschnittliche Ausdehnung in  $x_2$ -Richtung<sup>5</sup> signifikant kleiner. Ähnlich wie bei der Mutation erreicht die mittlere Annäherung die Größenordnung des Explorationsradius. Weiterhin trägt die Explorationsschrittweite entscheidend zu der durch die „Seitwärtsschritte“ erreichte Abdeckung der Paretomenge und Paretofront bei. Je größer die Schrittweite ist, desto weiter wandert der Suchmechanismus im kontradiktorischen Kegel entlang der Lösungsmenge. Exemplarisch können wir beide Effekte in den kumulativen Abbildungen 7.3 und 7.4 beobachten. Durch den größeren Explorationsradius streuen die gefundenen Zwischenlösungen in Abbildung 7.4 stärker um die optimale Lösungsmenge, decken jedoch auch einen größeren Teil der Menge und damit der Paretofront ab.

Die zusätzliche Betrachtung einer größeren Menge von Explorationsschritten ( $t = 10$ ) für die Gradientenapproximation trägt zu einer signifikanten Verringerung der Konvergenzgeschwindigkeit bei. Dabei ist der Gewinn in der Annäherungsgeschwindigkeit aber relativ gering im Vergleich zum Aufwand vieler zusätzlicher Funktionsauswertungen. Somit können wir feststellen, dass der Nachteil weniger

<sup>5</sup>Dieser Wert bildet das Mittel über die  $x_2$ -Komponenten aller gefundenen Lösungspunkte während eines Suchlaufes (kumulativ) und ist hier nicht explizit in Tabelle 7.1 aufgeführt.

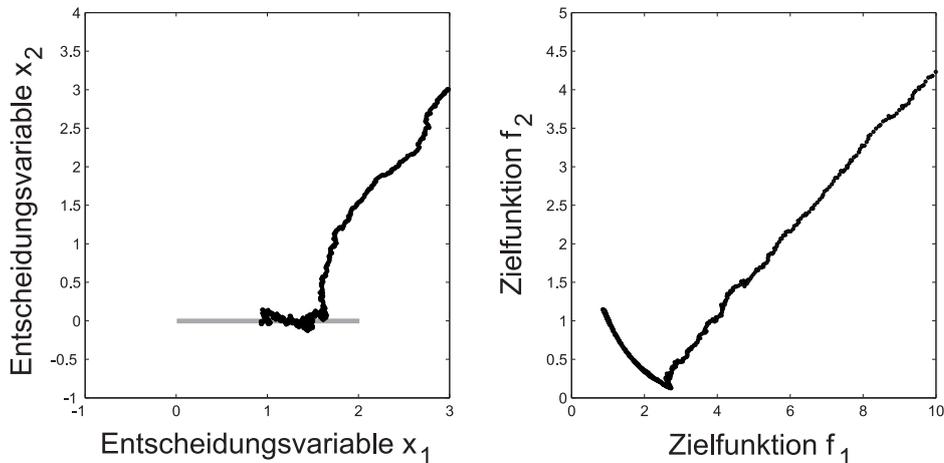


ABBILDUNG 7.3: Exemplarische Darstellung des Gradientenabstiegs für das Problem  $\mathcal{F}_{\odot}^2$  im Entscheidungs- und Bildraum ohne Räubereinfluss. Die Punkte stellen die kumulativen Ergebnisse von 2000 Iterationen dar. Die dargestellte Konfiguration geht von einem einzigen im Punkt  $(3, 3)$  initialisiertem Individuum und den Parametern  $t = 1$  und  $r = 0.01$  aus.

Explorationen zur Gradientenbildung zwar in der Länge der initialen Phase feststellbar aber gering ist. Die einfache Methode zur Gradientenapproximation scheint also ausreichend genau für unsere lokale Heuristik zu sein. Erst bei einer größeren Anzahl von Dimensionen ist die schwache Exploration nicht mehr uneingeschränkt einsetzbar. Exemplarisch sind im Anhang A.3 Ergebnisse für verschiedene Konfigurationen von  $t$  und  $r$  im 10-dimensionalen Entscheidungsraum des Problems  $\mathcal{F}_{\odot}^2$  dargestellt. Dort wird deutlich, dass die Konvergenzgeschwindigkeit der Methode stark abnimmt, wenn die Dimensionalität des Problems im Suchraum erhöht wird. Eine Erhöhung der Anzahl der Explorationsversuche  $t$  führt zu einer erheblichen Zeitersparnis bei der Annäherung an die Lösungsmenge.

Insgesamt zeigt die Untersuchung des Mechanismus, dass die gewünschten Eigenschaften grundsätzlich erfüllt werden. Die Methode sichert die Annäherung an die Lösungsmenge durch eine Fortbewegung in die kooperative Richtung aller Kriterien. In der Nähe der Lösungsmenge ist der Mechanismus in der Lage gleichsam automatisch in eine kontradiktorische „Seitwärtsbewegung“ umzuschalten und im Sinne der Lösungsdiversität entlang der optimalen Menge zu wandern. Das Umschalten zwischen dem Annäherungs- und Seitwärtsmodus geschieht dabei ohne eine explizite Parametrisierung oder die Angabe von Umschaltswellen. Ausschließlich die von Fritz John formulierte Eigenschaft des Gradientengleichgewichtes im lokal Pareto-optimalen Falle sorgt für das entsprechende Suchverhalten.

#### 7.4.2 Anwendung auf ein nicht-konvexes Problem

Nach der zuvor durchgeführten Analyse des lokalen Suchverfahrens soll nun dessen Integration über die Beuteindividuen in das Räuber-Beute-Modell genauer untersucht werden. Wir wählen dazu das zuvor bereits untersuchte Kursaweproblem,

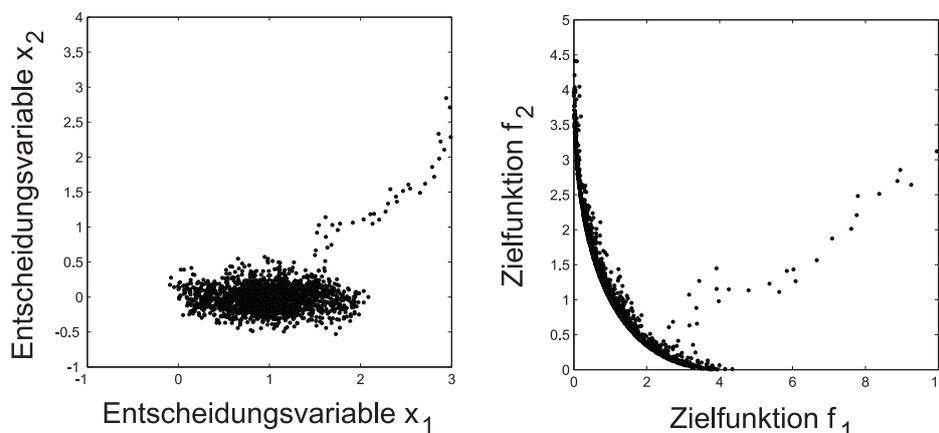


ABBILDUNG 7.4: Exemplarische Darstellung des Gradientenabstiegs für das Problem  $\mathcal{F}_{\odot}^2$  analog zur vorherigen Abbildung. Die Konfiguration unterscheidet sich ausschließlich im Parameter  $r = 0.1$ .

da es als eines der wenigen Testprobleme über eine nicht-konvexe Paretomenge und verschiedene lokal optimale Lösungen verfügt. Zudem ist es mit zwei Kriterien gut visualisierbar. Es sei hier nochmals explizit darauf hingewiesen, dass wir keinen Leistungsvergleich mit etablierten Verfahren der mehrkriteriellen Optimierung auf künstlichen Testproblemen anstreben, sondern vielmehr die Integration des entwickelten Suchverfahrens in das modulare Räuber-Beute-Modell und die Möglichkeiten der Verbesserung des existierenden Ansatzes beurteilen wollen. Zum Abschluss der Untersuchung verweisen wir auf entsprechende Referenzergebnisse.

Für die Untersuchung erlauben wir jeder Beute zu jeder Zeit, die durch sie repräsentierte Lösung selbstständig zu verbessern (wir bezeichnen diese Variante des Modells kurz als PPM-L). Dies kann durch die Aktivierung eines zum Gesamtmodell nebenläufigen Prozesses umgesetzt werden, indem wiederholt die in Abschnitt 7.2 definierte Vorschrift 7.3 angewendet wird. Die Parameter für die Gradientenapproximation wählen wir nach der vorhergehenden Analyse als  $t = 1$  und  $r = 0.01$ . Damit wollen wir einerseits die eingesetzten Funktionsauswertungen reduzieren und andererseits eine gute mittlere Annäherung an die Lösungsmenge erreichen. Wir erwarten, dass die relativ langsame Annäherung an die Paretomenge durch den globalen Einfluss der Räuber verkürzt wird und zugleich der kleine Explorationsradius zur Stabilisierung der Lösung beiträgt. Eine große kontradiktorische Bewegungsfähigkeit berücksichtigen wir mit geringerer Priorität, da nun 225 Individuen gleichzeitig suchen und insgesamt zu einer guten Abdeckung der Lösungsmenge führen sollten.

Für die Evaluation wählen wir 4 Räuber, wobei jeweils zwei Räuber ein gemeinsames Kriterium verfolgen. Diese Räuber unterteilen sich wiederum in einen die Gauss'sche Mutation ( $\sigma = 0.05$ ) als Operator tragenden Räuber und einen Räuber mit Simplexrekombination. Analog zum Versuchsaufbau der Vorstudie in Kapitel 6 wählen wir einen Torus der Größe  $15 \times 15$  mit 225 Beuteindividuen und lassen insgesamt 1000 Räuberbewegungen zu. Wir betrachten zusätzlich das

Räuber-Beute-Modell ohne integrierte lokale Suche (kurz als PPM bezeichnet) und vergleichen anschließend den Qualitätsgewinn durch den Mechanismus. Die Qualität der erreichten Approximation beurteilen wir entsprechend zweier üblicher Bewertungsmaße aus der Literatur. Beide stellen wir im Folgenden kurz vor.

**Das Hypervolumen (HV)** Zitzler (1999) hat in seiner Doktorarbeit vorgeschlagen, die Approximation einer Paretofront mittels des von dieser Approximation und einem Referenzpunkt eingeschlossene Volumen im Zielraum zu bewerten. Dieses sogenannte Hypervolumen wird zwischen den nicht dominierten Lösungen und dem Referenzpunkt entlang der Koordinatenachsen des Zielraumes gebildet. Formal ergibt sich das Hypervolumen einer Lösungsmenge  $\mathcal{L} = \{\ell_i \in \mathbb{R}^m \mid \ell_i, i = 1, \dots, \omega, \text{ Elemente aus Lösungsmenge } \tilde{\mathcal{Q}}\}$  mit  $\omega$  Lösungsvektoren für  $m$  Kriterien aus der Betrachtung aller Quadern  $Q_{\ell_i, R}$  für die Lösungsvektoren  $\ell_i$  mit dem Referenzpunkt  $R$  und entlang der Koordinatenachsen. Das Volumen  $\mathcal{V}$  der gesamten Lösungsmenge ergibt sich dann aus der Vereinigung aller Quader:

$$\mathcal{V} = \bigcup_{i=1}^{\omega} Q_{\ell_i, R} \quad (7.4)$$

Im Allgemeinen wird vorgeschlagen, die Lösungen entsprechend der Grenzen der ermittelten Lösungsmenge oder anhand gegebener Grenzen zu normieren, um eine gleichmäßige Gewichtung aller Kriterien sicherzustellen (Zitzler et al., 2003).

Das Hypervolumen hat als Maß zwei wichtige Eigenschaften. Es kann sowohl als Indikator für die Annäherung an die tatsächliche Paretofront wie auch als Maß für die Diversität der Lösung betrachtet werden. Gleichzeitig ist es in seiner Aussage unabhängig von dem Wissen über die tatsächliche Paretofront und dient somit oftmals als Grundlage für den Vergleich von Ergebnissen verschiedener Algorithmen, aber auch als Indikator zum Vergleich einzelner Lösungen innerhalb der Selektion mehrkriterieller Algorithmen (siehe etwa IBEA oder SMS-EMOA in Abschnitt 3.2.2). Es ist trotzdem zu beachten, dass das ermittelte Hypervolumen stark von der Wahl des Referenzpunktes und der Form der Paretofront abhängt. So können einige Punkte einen größeren Volumenbeitrag liefern als andere (Judt, Mersmann & Naujoks, 2012).

**Die Inverted Generational Distance (IGD)** Die Inverted Generational Distance ist eine von der ursprünglichen Generational Distance (Veldhuizen & Lamont, 1998) abgeleitete Modifikation. Mit ihr wird eine wichtige Schwäche des originalen Konzeptes behoben (Coello et al., 2007). Misst die Generational Distance ausgehend von der Front approximierter Lösungen die durchschnittliche Distanz zur Paretofront, so betrachtet die IGD den umgekehrten Weg. Damit wird verhindert, dass wenige und an einer Stelle gehäufte Lösungen ein gutes Ergebnis liefern, obwohl die gesamte Paretofront nicht abgedeckt wird.

Die Vorschrift der IGD beschreibt den Mittelwert aller euklidischer Distanzen

$d_i$  für  $k$  Elemente aus der Paretofront  $\mathcal{Q}^*$  zum nächsten Element der ermittelten Lösungsmenge  $\tilde{\mathcal{Q}}$ :

$$IGD = \frac{\sqrt{\sum_{i=1}^k d_i}}{|\mathcal{Q}^*|} \quad (7.5)$$

Es ist offensichtlich, dass dieses Maß nur anwendbar ist, wenn die Paretofront des betrachteten Problems bekannt ist.

Die Evaluation der Varianten PPM und PPM-L führen wir in einem ersten Experiment mit zufälliger Initialisierung im Bereich  $[-5, 5]^2$  durch. Für eine statistisch valide Aussage wiederholen wir die Experimente 30 mal und berechnen den Mittelwert und die Standardabweichung der ermittelten Maße. Um signifikante Unterschiede in den Ergebnissen nachzuweisen, verwenden wir wie zuvor den Wilcoxon Rangsummentest ( $p \leq 0.05$ ). Die Ergebnisse der Anwendung auf das Kursawe-Problem sind in Tabelle 7.2 zusammengefasst und in Abbildung 7.5 exemplarisch visualisiert. Als Referenzpunkt für die Auswertung des Hypervolumens nutzen wir den Punkt  $(-6, 0.5)$  und begrenzen die Betrachtung nach unten durch den Punkt  $(-10, -8)$  im Bildraum.

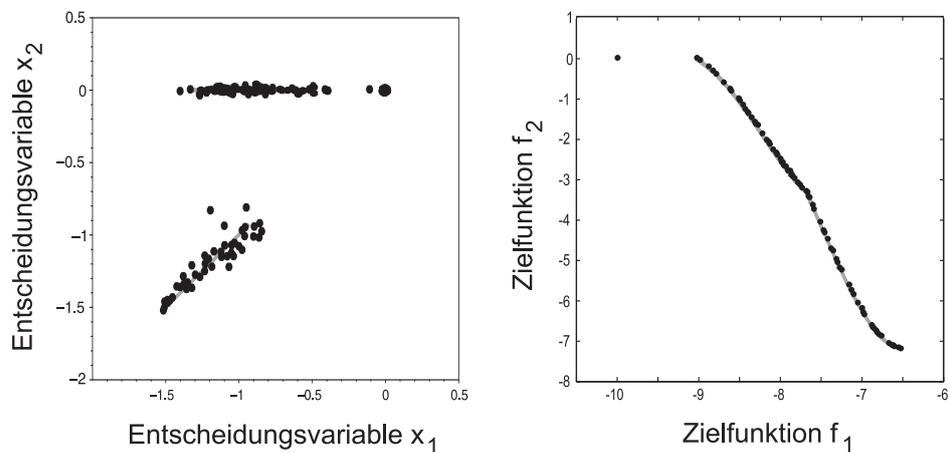


ABBILDUNG 7.5: Typische Lösung des Kursawe-Problems ( $n = 2$ ) für eine Evaluation mit integrierter gradientenbasierter Suche.

Offensichtlich ist das Räuber-Beute-Modell mit integrierter lokaler Suche nun in der Lage, die angestrebte Lösungsmenge an jeder Stelle gut zu approximieren. Alle Teile der Lösungsmenge, insbesondere jene mit lokalen Optima bezüglich der Kriterien, bleiben nun erhalten. Die Messwerte des Hypervolumens weisen eben-

TABELLE 7.2: *Ergebnisse der Auswertung des hybridisierten Modells mit Initialisierung in  $[-5, 5]^2$ , einer Mutationsschrittweite von  $\sigma = 0.05$ , und insgesamt 1000 Räuber-Läufen für das Kursawe-Problem. Es wurden 30 unabhängige Experimente pro Variante durchgeführt.*

Variante	HV	IGD
PPM	$0.401 \pm 0.0190$	$5.01 \cdot 10^{-4} \pm 3.84 \cdot 10^{-4}$
PPM-L	$0.415 \pm 0.0012$	$1.86 \cdot 10^{-4} \pm 3.71 \cdot 10^{-5}$

falls einen signifikanten Unterschied in der Lösungsqualität auf.<sup>6</sup> Ebenso ist die Annäherung durch das PPM-L an die Paretofront (siehe IGD) signifikant besser als die des PPM.

Wir können aus den Ergebnissen schließen, dass die Integration der lokalen Suche zu einer Stabilisierung der Lösungsmenge führt. Gleichzeitig müssen wir aber feststellen, dass durch den Einsatz der lokalen Suche eine große Anzahl an Funktionsauswertungen hinzugekommen ist. Zwar erlaubt der modulare Charakter des Räuber-Beute-Modells eine vollständig nebenläufige Ausführung, für einen fairen Vergleich über die Anzahl der Funktionsauswertungen müsste jedoch auch die Variante PPM mit der gleichen Anzahl von Auswertungen betrieben werden. Aus der Diskussion in Kapitel 6 ist aber klar, dass dies nicht zum Auffinden der vollständigen Paretomenge führte. Die nicht erweiterte Variante ist dazu grundsätzlich nicht in der Lage.

Durch die Erweiterung des Modells und die erhebliche Anzahl an zusätzlichen Funktionsauswertungen verändert sich aber auch die Gewichtung der einzelnen Aspekte des Modells. Insbesondere verliert die Räuberaktivität an Bedeutung bei der Lösungsformung. Dies wird durch die nun stark vertretene lokale Suche übernommen und führt zur Stabilisierung. Der Zweck der Räuberaktivität wird nun auf das Auffinden global optimaler Lösungen bezüglich der Kriterien, also auf das Auffinden des angestrebten Suchbereichs, reduziert. Diese wichtige Funktion kann der lokale Mechanismus nicht übernehmen. Wir untersuchen diesen Aspekt im Folgenden kurz.

Um den globalen Einfluss der Räuber darzustellen, betrachten wir das Kursawe-Problem genauer und initialisieren die zufällig erzeugte Population im Bereich  $[2, 12]^2$ . Innerhalb dieses Gebietes im Entscheidungsraum findet sich eine lokal optimale Lösungsmenge, die in den Abbildungen 7.6 und 7.7 dargestellt ist. Zusätzlich sind in beiden Abbildungen die global optimale Lösungsmenge und die Paretofront dargestellt.

Das Räuber-Beute-Modell ist in beiden Varianten (PPM und PPM-L) bei zu

---

<sup>6</sup>Der geringe Unterschied der absoluten Werte ist in der Nähe der vom Räuber-Beute-Modell gefundenen Lösungen im Bildraum begründet. Wie bereits in Abschnitt 6.3.2 und Abbildung 6.3 dargestellt, findet das unmodifizierte Räuber-Beute-Modell ausschließlich eine Lösungsmenge zwischen den Extremwerten  $a$  und  $b$ , die eine leicht zurückliegende Lösung im Bildraum erzeugt. Der Unterschied zur Paretofront ist jedoch gering und führt bei der Hypervolumenbetrachtung nur zu einem geringen Unterschied.

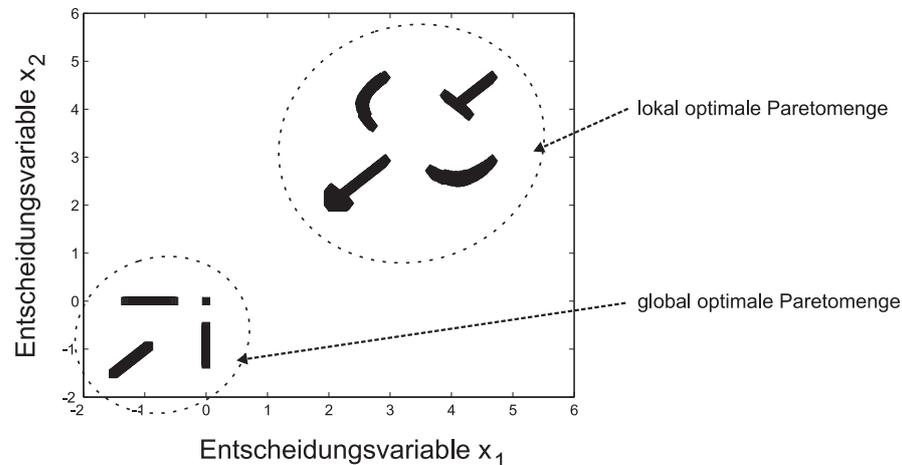


ABBILDUNG 7.6: Darstellung der globalen Paretomenge und einer lokal optimalen Paretomenge im Entscheidungsraum für das Kursawe-Problem. Die Mengen wurden auf Grundlage einer Rasterung (Rastergröße 0.01) des Entscheidungsraumes im Bereich  $[-5, 12]^2$  und durch anschließende Filterung der Lösungsmengen bestimmt.

klein gewählter Mutationsschrittweite ( $\sigma = 0.05$ ) nicht in der Lage diesen Initialisierungsbereich zu verlassen und bildet die lokale Paretofront aus. Wählen wir eine größere Mutationsschrittweite ( $\sigma = 0.5$ ), so wird der Initialisierungsbereich mit großer Wahrscheinlichkeit verlassen. Durch die Ausbildung stabiler Populationsgebiete setzen sich auf lange Sicht die gefundenen besseren Lösungen durch. In der Variante PPM-L schafft es dann die lokale Suche ebenfalls eine gute Approximation im Zielgebiet der global optimalen Paretomenge zu erzeugen. Tabelle 7.3 fasst die Ergebnisse der Auswertung mit abweichender Initialisierung und großer Mutationsschrittweite dar. Bereits nach 2000 Räuberbewegungen ist die Paretofront gut approximiert, signifikant besser als unter Einsatz der unveränderten Variante PPM.

TABELLE 7.3: Evaluation mit Initialisierung in  $[2, 12]^2$ , Mutation  $\sigma = 0.5$ , 2000 Räuber-Läufe und 30 unabhängige Experimente pro Variante.

Variante	HV	IGD
PPM	$0.399 \pm 0.0309$	$4.52 \cdot 10^{-4} \pm 5.93 \cdot 10^{-4}$
PPM-L	$0.403 \pm 0.0274$	$4.23 \cdot 10^{-4} \pm 6.71 \cdot 10^{-4}$

## 7.5 Diskussion der Erweiterung

Mit der Integration des entwickelten gradientenbasierten lokalen Suchverfahrens über das Modul der Beute ist es gelungen, das Räuber-Beute-Modell in der Lösungsbildung zu unterstützen. Durch die nebenläufige Durchführung der Suche, können wir das durch die Räuber induzierte globale Suchverfahren leicht von der

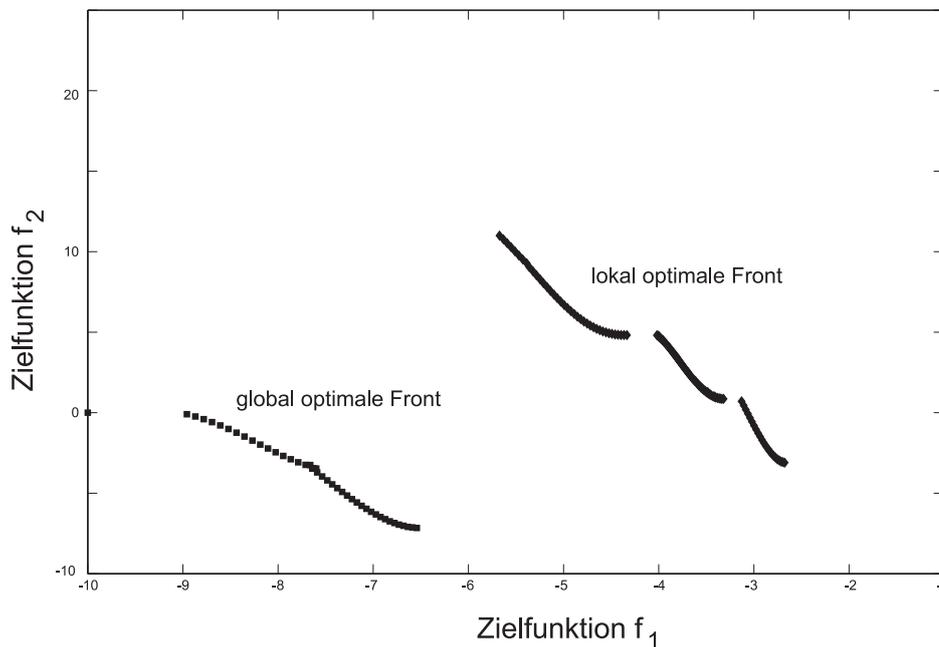


ABBILDUNG 7.7: Darstellung des Zielfunktionsraums des Kursawe-Problems. Hervorgehoben sind die globale Paretofront und eine lokal optimale Lösung, deren zugehörige Lösungsmenge im Bereich initialisierten Suchraumausschnittes  $[2, 12]^2$  liegt. Für die Erzeugung dieser Darstellung wurde eine Rasterung des Suchraums zwischen  $[-5, 12]^2$  ausgewertet. Die Rastergröße liegt bei 0.01.

Anwendung der lokalen Heuristik trennen und so das im Modell vorhandene Parallelisierungspotential gut nutzen. Außerdem haben wir anschaulich gezeigt, dass die modulare Struktur eine problemlose Erweiterung des Modells erlaubt, ohne gravierend in den algorithmischen Aufbau oder die Interaktion der Komponenten einzugreifen. Insgesamt stellt die Modifikation eine signifikante Verbesserung des Räuber-Beute-Modells dar und erweitert seine Anwendbarkeit auch auf Probleme mit nicht-konvexen Lösungsmengen.

Grundsätzlich ist aber im Kontext der evolutionären Algorithmen für mehrkriterielle Probleme festzustellen, dass diese Erweiterung das Räuber-Beute-Modell nicht konkurrenzfähig zu gebräuchlichen Ansätzen für das betrachtete Testproblem gemacht hat. Beispielhaft können die Ergebnisse von Auswertungen des Kursawe-Problems mit NSGA-2 und SMS-EMOA in Tabelle A.10 des Anhangs betrachtet werden. Zwar erreicht das Räuber-Beute-Modell für 1000 Räuberbewegungen eine bessere Approximationsqualität, insgesamt werden aber erheblich mehr, wenn auch nebenläufig ausgeführte, Funktionsauswertungen benötigt. Bei höheren Dimensionen des Entscheidungsraums werden sogar noch mehr Funktionsauswertungen benötigt, um eine gute Approximation des Gradienten zu gewährleisten.

Wir werden daher im nächsten Kapitel einen alternativen Ansatz zur Verbesserung des Räuber-Beute-Modells betrachten, der ebenfalls auf vorherigen Ergebnis-

sen der Dynamikanalyse in Kapitel 5 aufbaut und die  $\epsilon$ -Constraint-Methode (siehe Abschnitt 3.1.2) als eine zusätzliche Strategie zur Diversitätserhaltung einsetzt.

## Kapitel 8

# Diversitätserhaltung durch $\epsilon$ -Constraints

Der im vorherigen Kapitel betrachtete Ansatz erweitert das Räuber-Beute-Modell durch die Integration einer lokalen Suche in die Beute-Individuen. Obwohl dieser Ansatz in der Lage ist, nicht-konvexe Paretomengen zu approximieren und zudem die Modularität und Parallelität des Räuber-Beute-Modells nutzt, setzt er eine große Menge von Funktionsauswertungen ein, um den lokalen Suchmechanismus zu realisieren. Im Falle eines hochdimensionalen Entscheidungsraums steigt zusätzlich die Anzahl der Auswertungen für die genaue Approximation der Gradientenrichtung.

In diesem Kapitel soll nun ein alternativer Ansatz erprobt werden, der den Selektionsmechanismus innerhalb der Räuber verändert, um eine gute und diverse Approximation der Paretofront zu erreichen. Dabei wird die Fokussierung auf die Paretomenge aufgegeben. Die Paretofront rückt stärker in den Mittelpunkt. Zugleich wird ein kooperatives Verhalten zwischen den Räubern eingeführt, das beim Auffinden des angestrebten Lösungsgebietes hilft und eine Ausbildung von Nischen zur Betonung einzelner Abschnitte der Paretofront ermöglicht. Dabei nutzen wir die zuvor beobachtete und analysierte Tendenz des Räuber-Beute-Modells hin zu Extremlösungen. Zuerst diskutieren wir die grundlegende Idee der Integration eines der  $\epsilon$ -Constraint-Methode ähnlichen Ansatzes in die Räuberindividuen. Anschließend beschreiben wir detailliert die Umsetzung der Modifikationen und die für die angestrebte Nischenbildung notwendige, und durch die Kooperation von Räubern erreichte Exploration des Entscheidungsraums. Abschließend evaluieren wir den Ansatz mit dem nicht-konvexen Kursawe-Problem und den im Entscheidungsraum hochdimensionalen ZDT-Problemen.

### 8.1 Integration von Zielbeschränkungen

Die Analyse der Dynamik des Räuber-Beute-Modells in Kapitel 5 begründet, warum der Ansatz eine starke Tendenz in die globalen Extremlösungen eines mehr-

kriteriellen Problems besitzt. Im vorhergehenden Kapitel haben wir versucht, diese Eigenschaft durch die Integration einer starken lokalen Suche zu kompensieren. Folgt man aber dem Ansatz von Grimme & Lepping (2011b), so kann man diese Eigenschaft, im Rahmen der Selektion und kombiniert mit einer neu eingeführten Interaktion der Räuber, zur Erzeugung einer diversen Paretofront *nutzen*. Dazu ist es nötig, dass die Lösungen von den Räubern nicht ausschließlich zu den Extrempunkten der Paretofront getrieben werden, sondern entlang der gesamten Front entstehen können.

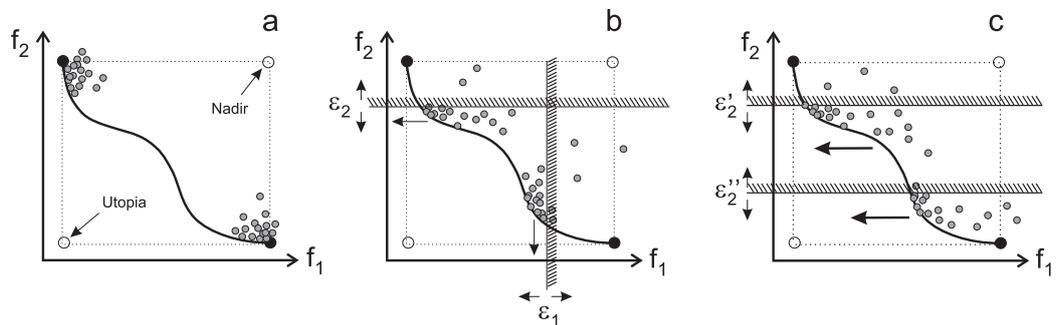


ABBILDUNG 8.1: Schematische Darstellung der grundlegenden Idee der  $\epsilon$ -Constraint-Integration. Die Räuber bestimmen kooperativ den Utopia- und Nadirpunkt mit dem Ziel ein Gebiet festzulegen, in dem die Grenzen angewendet werden, siehe (a). Um verschiedene Teile der Paretofront zu betonen, wird jeder Räuber Grenzen für die nicht von ihm verfolgten Kriterien ausgestattet, siehe (b). Durch individuelle Einstellung der Grenzen – in (c) nur für Räuber  $f_2$  dargestellt – wird erwartet, dass die Beutepopulation an den Grenzen stabile Subpopulationen ausbilden. Die Pfeile in (b) und (c) stellen die Entwicklungstendenz der vom Räuber lokal beeinflussten Subpopulation dar.

Um dies zu erreichen, bietet sich die in Abschnitt 3.1.2 beschriebene  $\epsilon$ -Constraint-Methode an. Analog zum Vorgehen im Räuber-Beute-Modell wird dort nur ein einziges Kriterium betrachtet. Für alle anderen Kriterien werden obere Wertegrenzen definiert, die bei der Optimierung des Hauptziels nicht überschritten werden dürfen. Der Ansatz zur Übertragung dieses Prinzips bedeutet im Räuber-Beute-Modell eine Modifikation des Räubers, also der Selektionskomponente selbst. Dabei selektiert jeder Räuber wie bisher nach einem einzigen Kriterium. Jedoch geht in seine Qualitätsbewertung einer Lösung, zusätzlich zum Funktionswert, die Verletzung von Grenzen bezüglich aller vom Räuber nicht betrachteten Kriterien ein. Damit wird das grundlegende Verhalten des Räubers nicht verändert, es wird aber ein Mechanismus zur Begrenzung der Extremalentwicklung eingeführt. Der Räuber verändert durch die zusätzliche Berücksichtigung von Obergrenzen für  $m - 1$  Kriterien implizit sein Zielgebiet auf der Paretofront. Abbildung 8.1 zeigt in seinen Teilbildern (a) und (b) genau diese Veränderung: Exi-

stieren in (a) keine Grenzen, so tendieren die Lösungen unter dem ausschließlichen Einsatz von Mutation<sup>1</sup> in die globalen Extrempunkte der Paretofront.

Fügt man, wie in Bild (b) der Abbildung 8.1 dargestellt, für jeden Räuber eine Grenze hinzu, so tendieren die Extrem Lösungen gegen die Grenzen des jeweils nicht vom Räuber verfolgten Kriteriums. In der Abbildung wird dem Räuber mit Ziel  $f_1$  die Grenze  $\varepsilon_2$  für das Kriterium  $f_2$  und dem Räuber mit Ziel  $f_2$  die Grenze  $\varepsilon_1$  für das Kriterium  $f_1$  hinzugefügt. Der Pfeil, ausgehend von den jeweiligen Lösungspunkten, gibt die Tendenz der Individuenentwicklung unter dem Einfluss des Räubers und seines zugeordneten Kriteriums an. Diese Richtung ist jedoch durch die Grenzen des jeweils anderen Kriteriums weitgehend beschränkt. Nun können wir erwarten, dass durch eine sukzessive Veränderung der Grenzen die Extremwertbildung entlang der Paretofront verschoben werden kann. Mit einer vom Algorithmus unabhängigen Archivierung können so bereits erreichte Lösungen gespeichert werden. Sie ergeben schließlich eine diverse Lösungsfront. Gleichzeitig können wir erwarten, dass mehrere Räuber gleicher Art, aber mit unterschiedlichen Grenzen (siehe dazu eine vereinfachte Darstellung bezüglich Kriterium  $f_1$  in Abbildung 8.1 (c) die Erzeugung unterschiedlicher Gebiete an der Paretofront beschleunigen.

Voraussetzung für die Wahl geeigneter Grenzen bezüglich der Kriterien und für die Variation eben dieser Grenzen im Zuge der algorithmischen Ausführung, ist die Bestimmung des angestrebten Zielbereichs. Dieser kann allgemein durch zwei besondere Punkte im Lösungsraum definiert werden: durch den Utopia- oder Idealpunkt und den sogenannten Nadirpunkt. Der Utopiapunkt ist ein Vektor  $\vec{u} = (u_1, \dots, u_m)^T$ ,  $u_i \in \mathbb{R}$  und beschreibt eine hypothetische Position im Suchraum. Jede Komponente  $u_i$ ,  $i = 1, \dots, m$  stellt dabei die optimale Lösung des Kriteriums  $i$  dar und repräsentiert somit eine Ideallösung unter der Annahme keines Widerspruchs unter den Kriterien. Der Nadirpunkt (*nadir*: arab. Fußpunkt, gegensätzlich zum Utopiapunkt) beschreibt den gegenteiligen Punkt und ist erheblich schwieriger zu bestimmen. Der Vektor  $\vec{\eta} = (\eta_1, \dots, \eta_m)^T$ ,  $\eta_i \in \mathbb{R}$ , kann für jede Komponente  $\eta_i$  über das Maximum derjenigen Werte bestimmt werden, die das Kriterium  $i$  annimmt, wenn jedes andere Kriterium optimal gewählt wird. Mit anderen Worten kann für jedes Kriterium  $j \neq i$  ein minimaler Wert  $f_i$  unter der Voraussetzung  $f_j = opt$  bestimmt werden: Der sogenannte „Pay-off“ der Lösung von  $f_i$  unter Optimalität von  $f_j$ . Das Maximum aller Pay-offs für Kriterium  $i$  ergibt eine obere Schranke für das Kriterium und die entsprechende Komponente des Nadirvektors. Der von Utopia- und Nadirpunkt eingeschlossene Lösungsraum ist jenes Gebiet, das für die Bestimmung der Grenzen der Räuber ausschlaggebend ist. Wir werden später genauer diskutieren, wie die Kooperation von Räubern zur automatischen Bestimmung dieses Gebietes beiträgt. Im Folgenden gehen wir nun auf die algorithmische Umsetzung des Ansatzes ein.

<sup>1</sup>Wir betrachten zur einfacheren Beschreibung der erwarteten Dynamik zuerst den Fall ausschließlicher Mutation, wie dies bereits Grimme & Lepping (2011b) in ihrer Arbeit getan haben. In unseren später durchgeführten Experimenten fügen wir die Simplexrekombination als den die Konvergenz unterstützenden Operator wieder hinzu.

## 8.2 Algorithmische Erweiterungen und Implementierung

Für die Umsetzung des oben beschriebenen Konzeptes selektiert jeder Räuber weiterhin nach seinem zugeordneten Kriterium  $f_c \in \mathcal{F}, c \in \{1, \dots, m\}$ , berücksichtigt dabei aber zusätzlich alle  $m - 1$  weiteren Kriterien als Grenzen. Diese werden formal durch einen für jeden Räuber individuellen Vektor  $\vec{\beta} = (\beta_1, \dots, \beta_m)^T, \beta_i \in \mathbb{R}, i = 1, \dots, m$  mit  $\beta_c = \infty$  ausgedrückt. Die Integration in das allgemeine Räuberverhalten aus Algorithmus 1 erfolgt direkt über den Selektionsmechanismus und ist in Algorithmus 2 dargestellt.

---

**Algorithmus 2** Allgemeines Räuberverhalten.

---

**Eingabe:**  $\mathfrak{R} \ni \mathfrak{r} = (f_c, op), v_0, r, \pi_0, \vec{\beta}$

- 1: **while** NOT (Terminierungskriterium erfüllt) **do**
  - 2: Zufallslauf  $\pi_i \rightarrow \pi_{i+1}$
  - 3:  $SN = RN = \mathfrak{R}(\pi_{i+1}, r)$
  - 4: bewerte Beute in  $SN$  bzgl.  $f_c$  und  $\vec{\beta}$ , markiere schlechteste Beute  $b_{worst} \in \mathfrak{B}_{SN}$  zur Ersetzung
  - 5: Nachkomme  $b_{off} := op(RN \setminus b_{worst})$
  - 6: ersetze  $b_{worst}$  durch  $b_{off}$
  - 7: **end while**
- 

Konkret geschieht die Integration in Zeile 4. Hier wird die Bewertung um die Berücksichtigung des Grenzenvektors  $\vec{\beta}$  erweitert. Danach wird die Räuberaktion normal fortgesetzt. Da sich alle Räuber mit paarweise unterschiedlichen Grenzen auf einer räumlich sehr begrenzten Populationsstruktur bewegen, wird der elitäre Charakter der Ersetzung des schlechtesten Individuums hier aufgegeben, siehe Zeile 6. Eine neue Lösung ersetzt hier grundsätzlich und unabhängig von der Qualität des Nachkommens das schlechteste Individuum.

---

**Algorithmus 3**  $\varepsilon$ -Constraint Bewertung.

---

**Eingabe:**  $bewerte = (\mathbf{b}, f_c, \beta, min_{feas}, max_{feas})$

- 1:  $strafe = 0$
  - 2:  $f = \mathbf{b}.fitness$
  - 3: **for all**  $f_i \in f$  with  $i \neq c$  and  $\beta_i < f_i$  **do**
  - 4:  $intervall(i) = max_{feas}(i) - min_{feas}(i)$
  - 5:  $verletzung(i) = (f_i - \beta_i) / intervall(i)$
  - 6:  $strafe = strafe + verletzung(i) \cdot \omega(i)$
  - 7: **end for**
  - 8: return  $f_c + \exp(\max(0, strafe)) - 1$
- 

Das Vorgehen für die modifizierte Bewertung eines Individuums durch den Räuber ist in Algorithmus 3 dargestellt. Hier wird für die Qualitätsbewertung ne-

ben dem Funktionswert die Verletzung der Obergrenzen für alle übrigen  $m - 1$  Kriterien berücksichtigt. Eingaben für diese Bewertung sind die zu bewertende Beute  $\mathbf{b}$ , die vom Räuber verfolgte Zielfunktion  $f_c$ , der Grenzenvektor  $\vec{\beta}$  und die momentan ermittelten Grenzen  $\min_{feas}$  und  $\max_{feas}$  des angestrebten Zielraums. Die Bestimmung der Zielraumgrenzen durch eine Approximation von Utopia- und Nadirpunkten wird im anschließenden Abschnitt ausführlich betrachtet. Hier wird davon ausgegangen, dass diese Grenzen  $\min_{feas}$  und  $\max_{feas}$  bekannt sind. Die zum Funktionswert hinzugerechnete Strafe (siehe Zeile 6) ergibt sich durch komponentenweise Betrachtung des Grenzenvektors und des zugehörigen ermittelten Kriteriumswertes. Die bei der Berechnung verwendete Verletzung der Grenze wird bezüglich der entsprechenden Intervalllänge in der betrachteten Komponente des angestrebten Zielraums normiert. Wird die Grenze verletzt, so ist der Verletzungswert positiv und muss als gewichtet in der Gesamtstrafe berücksichtigt werden. Die Gewichtung geschieht entsprechend einer vom Kriterium abhängigen Gewichtungsfunktion  $\omega : \{1, \dots, m\} \rightarrow \mathbb{R}$ , die den jeweiligen Wertebereich des Kriteriums berücksichtigt. Für unseren Fall betrachten wir vereinfachend  $\omega(i) = \max_{feas}(i)$ . Somit beziehen wir die aktuelle Ausdehnung des angestrebten Bereiches, ausgehend vom Koordinatenursprung, in die Gewichtung ein. Es ist zu beachten, dass diese Vereinfachung der Gewichtung nur wie gewünscht funktioniert, wenn die obere Grenze des angestrebten Bereiches im ersten Orthanten<sup>2</sup> des betrachteten Suchraum liegt. Für unsere hier verwendeten Probleme können wir diese Eigenschaft durch Verschiebung im Bildraum erreichen. Die so ermittelte Gesamtstrafe für alle Kriterien wird als Exponent innerhalb einer einfachen Exponentialfunktion verwendet, die schließlich die tatsächliche Erhöhung des Funktionswertes bestimmt. Eine Exponentialfunktion wird hier genutzt, um kleine Überschreitungen der Grenzen weniger stark als große Überschreitungen der Grenzen zu bestrafen. Dadurch wird innerhalb enger Grenzen weiterhin ein exploratives Verhalten jenseits der bisher ermittelten Grenzen zur potentiellen Verbesserung der Utopia-Informationen ermöglicht.

### 8.2.1 Automatisches Auffinden des Zielgebiets im Entscheidungsraum

Algorithmus 3 zur Bewertung der Lösungsqualität unter Berücksichtigung der  $\epsilon$ -Constraints setzt voraus, dass die Vektoren  $\min_{feas}$  und  $\max_{feas}$  existieren, die das gewünschte Zielgebiet im Bildraum definieren. Initial sind diese Werte für unseren Ansatz nicht verfügbar, so dass auch grundsätzlich keine Begrenzungen für die Räuber im Bildraum definiert werden können. Um dieses Problem zu beheben, können wir zwei Ansätze betrachten: Einerseits kann der Nutzer des Räuber-Beute-Modells initiale Grenzen festlegen, um die Suche auf ein von ihm angestrebtes Gebiet im Bildraum zu beschränken. Dies setzt voraus, dass er in der Lage ist, die zu erreichende Lösung und deren Lage im Bildraum abzuschätzen. Andererseits,

<sup>2</sup>Der Begriff des *Orthanten* ist die  $n$ -dimensionale Verallgemeinerung der Begriffe des Quadranten im  $\mathbb{R}^2$  und des Oktanten im  $\mathbb{R}^3$ .

wenn dies nicht möglich ist, muss es einen Mechanismus im Algorithmus selbst geben, der in der Lage ist aus den ermittelten Lösungen den gewünschten Zielbereich sukzessive zu ermitteln. Dies bedeutet, dass Utopia- und Nadirpunkt approximiert werden müssen, da beide eine untere bzw. obere Grenze für die Ausdehnung der Paretofront darstellen. Die starke Tendenz der Räuber Extrem Lösungen auszubilden, wollen wir im Folgenden nutzen, um einen einfachen Mechanismus zu entwerfen, der genau diese automatische Detektion des Zielgebietes mittels Approximation von Utopia- und Nadirpunkt ermöglicht.

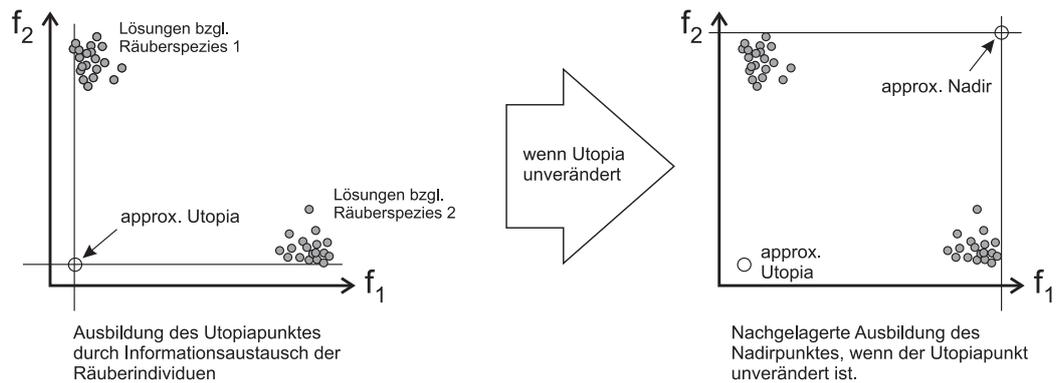


ABBILDUNG 8.2: Ausbildung der Utopia- und Nadirpunkte als zweistufiger Prozess. Durch den Austausch von Zielfunktionswerten bzgl. der individuell verfolgten Kriterien zwischen den Räubern ist jeder Räuber mittelfristig im Besitz vollständiger Utopia-Information. Nachgelagert werden bei gleicher Utopiainformation zwischen den Räubern Informationen bezüglich der Funktionswerte aller nicht verfolgten Kriterien ausgetauscht. Daraus können die Nadirinformationen gebildet werden.

Der Prozess der Ermittlung der beiden maßgeblichen Punkte ist zweistufig und ist in Abbildung 8.2 dargestellt. Mittels Räuberinteraktion wird der Utopiapunkt bestimmt. Nur wenn diese Information ausreichend exakt ermittelt wurde, ist der zugehörige Nadirpunkt als obere Grenze für das gewünschte Zielgebiet aussagekräftig. Daher wird der Nadirpunkt erst stabil ausgebildet, wenn der Utopiapunkt weitgehend unverändert bleibt. Somit ist die Ausbildung des Utopiapunktes eine Voraussetzung für die stabile Entwicklung der Nadirinformation. Die Ermittlung beider Informationen wird im Folgenden beschrieben.

Jeder Räuber beginnt seine Suche mit unspezifizierten Utopia- und Nadirinformationen. Erst im Laufe der Evolution bilden sich diese Punkte heraus. Dazu speichert jeder Räuber den bezüglich seines eigenen Kriteriums bisher besten erreichten Funktionswert, quasi eine individuelle Abschätzung der Utopiakomponente seines Kriteriums. Treffen nun zwei Räuber auf der gleichen Torusposition zusammen, so tauschen sie alle gesammelten Informationen (natürlich bzgl. aller Kriterien) aus und vervollständigen so ihre jeweilige Utopiainformation. Durch das wiederholte Aufeinandertreffen der Räuber ergibt sich so sukzessive eine Approximation des Utopiapunktes. Wir nutzen also die Tendenz in die Extremwerte der Paretofront in ihrer Kombination über die Räuber als quasi parallelen Approximationsansatz

für den Utopiapunkt. Da wir initial keine Grenzen für die Räuber festsetzen, also  $\beta_i = \infty, i = 1, \dots, m$  fordern, erwarten wir, dass sich der Utopiapunkt gut herausbilden kann.

Die Approximation des Nadirpunktes ist komplexer und wird aufbauend auf und nachgelagert zur Ermittlung der Utopia-Information durchgeführt. Zwar wird die Information auch zwischen zwei Räubern auf der gleichen Position des Torus ausgetauscht, jedoch wird immer das Maximum der jeweiligen Komponenten konserviert. Die Lebensdauer des ermittelten Nadirpunktes eines Räubers hängt von der Stabilität des im Räuber gespeicherten Utopiapunktes ab. Verändert sich der Utopiavektor, so wird die ermittelte Nadir-Information gelöscht und muss anschließend durch Begegnung mit anderen Räubern neu gelernt werden. Dies ist darin begründet, dass eine Verbesserung bezüglich des vom Räuber verfolgten Kriteriums zu einer Verschlechterung aller übrigen Kriterien unter dieser Bedingung führen kann. Die Konservierung der zuvor erlernten Nadirinformation würde damit zu einer Unterschätzung des Nadirpunktes und nur zu einer teilweisen Abdeckung des angestrebten Bildraumes führen. Das „Vergessen“ des approximierten Nadirpunktes ermöglicht so die Neubildung aktueller Informationen.

Insgesamt erwarten wir, dass durch das Extremwertverhalten der Räuber und durch die direkte Kooperation zum Informationsaustausch eine gute Approximation des Utopiapunktes und des Nadirpunktes erreicht wird. Beide Punkte bilden die Grundlage für die Bewertung gemäß Algorithmus 3 und definieren zugleich einen Bereich im Zielraum, in dem die individuellen Grenzen für die Räuber eingestellt werden können.

### 8.2.2 Nischenbildung durch Räubereinfluss

Mithilfe der zuvor ermittelten Informationen über das betrachtete Zielgebiet der Paretofront ist es nun möglich, einen einfachen Mechanismus zur Abdeckung der Paretofront durch individuelle  $\epsilon$ -Constraints der Räuber zu erreichen. Der Grenzenvektor  $\vec{\beta}$  eines jeden Räubers wird individuell angepasst. Wie bereits erwähnt, beginnt jeder Räuber seine Suche ohne Grenzen, um gemeinsam mit anderen Räubern die Ausbildung von Extremwerten zu ermöglichen und damit den Utopia- und den Nadirpunkt zu approximieren. Wenn schließlich die Utopia- und Nadirinformation verfügbar sind – also stabile Extrema der Paretofront durch die Räuber bestimmt sind – passen die Räuber ihre individuellen Grenzen für die nicht von ihnen verfolgten  $m - 1$  Kriterien innerhalb dieser unteren und oberen Grenzen an.

Initial wird die Grenze zufällig im Intervall der entsprechenden Komponenten von Utopia- und Nadirpunkt gewählt. Danach werden die Grenzen aktualisiert, wenn sich zwei Räuber *gleichen Kriteriums* auf dem Torus treffen, also dieselbe

Position besetzen.<sup>3</sup> Dabei wird eine normalverteilte Störung  $pert$  entsprechend Vorschrift (8.1)

$$pert = \mathcal{N}\left(0, \frac{\max(\beta_i - u_i, \eta_i - \beta_i)}{(\eta_i - u_i)}\right) \quad (8.1)$$

auf die bisherige Grenze des Räubers angewendet. Dies führt meist zu einer moderaten Verschiebung der individuellen Grenze und verändert damit das Selektionsverhalten des Räubers bezüglich des betroffenen Kriteriums; der Räuber wird entweder toleranter gegen die Erhöhung des Funktionswertes für dieses Kriterium oder aber restriktiver. Anders als durch eine gleichverteilte Wahl der neuen Grenze, kommt es durch die häufig kleinen Störungen durch eine normalverteilte Wahl nicht zu einer abrupten Veränderung des Selektionsverhaltens. Dies erlaubt es der Beutepopulation, schneller auf die Veränderungen durch geringe Anpassungen zu reagieren und damit rasch eine neue Position auf der Paretofront zu erreichen. Zudem erlaubt die normalverteilte Störung in seltenen Fällen ein Überschreiten der durch die approximierten Utopia- und Nadirpunkte festgelegten Grenzen des abgedeckten Bildraumes. Dies schafft für die Räuber die Möglichkeit, nicht nur im Sinne der Diversität Lösungen auf der Paretofront zu generieren, sondern weiterhin explorativ tätig zu sein, wenn der eigentliche Zielbereich noch nicht erreicht ist.

Technisch wird zur Konfiguration der Grenzanpassung die Standardabweichung der Störung gemäß Gleichung 8.1 aus der größten Entfernung der aktuellen Grenze  $\beta_i$  zu den Gebietsbegrenzungen – durch die entsprechende Utopiakomponente  $u_i$  oder Nadirkomponente  $\eta_i$  – normiert auf die Gebietsgröße bestimmt. Dieser Ansatz benötigt weder Informationen aus dem Evolutionsprozess noch Informationen, die durch andere Räuber bereitgestellt werden.

### 8.3 Evaluation des Ansatzes

Für die Bewertung des entwickelten Ansatzes betrachten wir zwei Arten von Problemen. Einerseits interessiert uns, ob auch dieser Ansatz in der Lage ist, nicht konvexe Probleme gut zu lösen. Dazu betrachten wir erneut das Kursawe-Problem. Damit die Anwendung der in Algorithmus 3 dargestellten Bewertungsmethode möglich ist, verschieben wir das Bild des Kursawe-Problems im Zielraum so, dass der Nadirpunkt oberhalb des Ursprungs liegt. Nach der Ausführung unseres Ansatzes machen wir die Verschiebung auf der ermittelten Lösungsmenge rückgängig. Die Problemkomplexität wird dadurch nicht verändert.

---

<sup>3</sup>Der Mechanismus der Grenzanpassung ist getrennt von der Approximation der Utopia- und Nadirinformationen zu betrachten. Er wird nur aktiviert, wenn zwei Räuber gleichen Kriteriums aufeinandertreffen. Der Approximationsvorgang von Utopia- und Nadirpunkten ist davon nicht berührt und wird unabhängig davon ebenfalls ausgeführt. Bei der Grenzanpassung findet kein Informationsaustausch statt.

Andererseits wollen wir überprüfen, inwieweit dieser Ansatz im Kontext hochdimensionaler Entscheidungsräume einsetzbar ist. Da der Ansatz im Bildraum arbeitet, ist er grundsätzlich unabhängig von der Dimension des Entscheidungsraumes einsetzbar. Jenseits des zu erwartenden langsameren Konvergenzverhaltens des Evolutionsprozesses – impliziert durch die Betrachtung vieler Dimensionen – wollen wir überprüfen, ob die Methode zu einer Steigerung der Ergebnisqualität beiträgt. Für diesen Teil der Evaluation betrachten wir die Probleme ZDT1 bis ZDT3 in einem 30-dimensionalen Entscheidungsraum genauer.

Im Falle des Kursawe-Problems werden insgesamt 5000 Funktionsauswertungen pro Experiment zugelassen, während aufgrund der höheren Dimensionalität der Probleme für die Auswertung von ZDT 1 bis 3 eine Anzahl von 10.000 Funktionsauswertungen erlaubt wird. Um eine Einordnung der Leistungsfähigkeit des Ansatzes (wir kürzen diesen im Folgenden mit  $\epsilon$ -PPM ab) vorzunehmen, greifen wir in diesem Falle auf zwei Referenzalgorithmen, NSGA-2 und SMS-EMOA, in ihrer Standardkonfiguration entsprechend Deb et al. (2002) sowie Emmerich et al. (2005) zurück. Zusätzlich zeigen wir die Auswertungsergebnisse für das unveränderte Räuber-Beute-Modell, um den Vorteil der eingesetzten Methode zu verdeutlichen. In unserem Ansatz verwenden wir 10 Räuber für alle Testprobleme auf einem Torus mit  $10 \times 10 = 100$  Individuen. Die relativ kleine Wahl des Torus stellt sicher, dass sich Räuber häufig begegnen und Informationen austauschen können. Die genaue Konfiguration der Räuber ist der Tabelle A.11 in Anhang A.5 zu entnehmen. Grundsätzlich werden für jeden Algorithmus 30 unabhängige Experimente mit zufälliger Initialisierung durchgeführt. Die Mittelwerte und Standardabweichungen des Hypervolumens (HV) und der Inverted Generational Distance (IGD) sind in Tabelle 8.1 angegeben. Im Folgenden getroffene Signifikanzaussagen über Vergleiche von Ansätzen auf Grundlage der Werte in Tabelle 8.1 sind mit dem Wilcoxon-Test ( $p \leq 0.05$ ) überprüft worden. Die Untersuchung des Kursawe-Problems mit dem  $\epsilon$ -PPM-Ansatz zeigt, dass der Ansatz zwar in der Lage ist, eine relativ gute Approximation der Lösungsmenge zu erstellen, jedoch nach 5000 Funktionsauswertungen noch nicht die Qualität von NSGA-2 bzw. SMS-EMOA erreichen kann. Aus Untersuchungen von Grimme & Lepping (2011b) mit über 50.000 Funktionsauswertungen ist jedoch festzustellen, dass bei einer größeren Anzahl von Funktionsauswertungen das  $\epsilon$ -PPM in der Lage ist, eine bessere Approximation als NSGA-2 zu erreichen. Offensichtlich benötigt der integrierte Mechanismus eine lange Anlaufzeit zur Etablierung verschiedener Nischen und damit zur Abdeckung der Paretofront. Dies wird auch durch den Vergleich des Ansatzes mit dem unveränderten Räuber-Beute-Modell (PPM) deutlich. Die Qualität der erreichten Approximation des  $\epsilon$ -PPM ist zwar signifikant größer als jene des PPM, jedoch ist der Unterschied nicht sehr ausgeprägt. Der Mechanismus zur Nischenbildung war also noch nicht lange aktiv. Diese Verzögerung liegt sicherlich in der initialen Auffindphase des angestrebten Zielgebietes, also der Utopia- und Nadirpunkte, begründet. Erst wenn diese gut approximiert sind, kann der Mechanismus zur Diversitätserhaltung seine Wirkung vollständig entfalten.

TABELLE 8.1: Auswertungsergebnisse für das Räuber-Beute-Modell mit Erweiterung um  $\epsilon$ -Constraints ( $\epsilon$ -PPM) im Vergleich zu Ergebnissen des unveränderten Modells (PPM), NSGA-2 und SMS-EMOA. Als Testprobleme wurden Kursawe mit  $n = 2$  und ZDT1-3 mit  $n = 30$  gewählt und jeweils 30 Experimente pro Algorithmus und Testproblem durchgeführt. Die angegebenen Werte für Hypervolumen (HV) und Inverted Generational Distance (IGD) sind Mittelwerte  $\pm$  Standardabweichung. Referenzpunkte für die HV-Berechnung von Ergebnissen für das Kursawe-Problem sind  $(-10, -8)$  und  $(-6, 0.5)$ . Für ZDT1 und 2 werden die Referenzpunkte  $(0, 0)$  und  $(1, 1)$ , für ZDT3  $(0, -0.8)$  und  $(1, 1)$  betrachtet.

Problem	Methode	HV	IGD
Kursawe	PPM	$0.406 \pm 0.0149$	$4.11 \cdot 10^{-4} \pm 2.15 \cdot 10^{-4}$
	$\epsilon$ -PPM	$0.408 \pm 0.0288$	$3.65 \cdot 10^{-4} \pm 5.99 \cdot 10^{-4}$
	NSGA-2	$0.417 \pm 3.91 \cdot 10^{-4}$	$1.13 \cdot 10^{-4} \pm 9.52 \cdot 10^{-6}$
	SMS-EMOA	$0.419 \pm 3.63 \cdot 10^{-4}$	$8.36 \cdot 10^{-5} \pm 2.81 \cdot 10^{-6}$
ZDT1	PPM	$0.540 \pm 0.0274$	$0.0025 \pm 5.78 \cdot 10^{-4}$
	$\epsilon$ -PPM	$0.607 \pm 0.0405$	$0.0013 \pm 9.28 \cdot 10^{-4}$
	NSGA-2	$0.593 \pm 0.0168$	$0.0015 \pm 3.64 \cdot 10^{-4}$
	SMS-EMOA	$0.618 \pm 0.0130$	$0.0477 \pm 7.57 \cdot 10^{-6}$
ZDT2	PPM	$0.193 \pm 0.0459$	$0.0046 \pm 0.0022$
	$\epsilon$ -PPM	$0.265 \pm 0.0329$	$0.0022 \pm 0.0018$
	NSGA-2	$0.210 \pm 0.0451$	$0.0037 \pm 0.0041$
	SMS-EMOA	$0.127 \pm 0.1190$	$0.0117 \pm 0.0088$
ZDT3	PPM	$0.500 \pm 0.0141$	$0.0021 \pm 5.46 \cdot 10^{-4}$
	$\epsilon$ -PPM	$0.529 \pm 0.0110$	$0.0014 \pm 4.67 \cdot 10^{-4}$
	NSGA-2	$0.527 \pm 0.0121$	$0.0011 \pm 4.27 \cdot 10^{-4}$
	SMS-EMOA	$0.542 \pm 0.0142$	$9.25 \cdot 10^{-4} \pm 6.96 \cdot 10^{-4}$

Die Ergebnisse für die hochdimensionalen Problemstellungen ZDT 1 bis 3 zeigen ebenfalls, dass eine häufigere Anwendung des  $\epsilon$ -Constraint-Mechanismus zu guten Ergebnissen führt. In allen Fällen erzeugt  $\epsilon$ -PPM signifikant bessere Hypervolumenergebnisse als die unveränderte Variante PPM. Zugleich ist der entwickelte Mechanismus in der Lage, vergleichbare Lösungsqualität zu NSGA-2 und SMS-EMOA zu liefern, sie teilweise sogar zu übertreffen. In den Fällen von ZDT1 und ZDT3 sind zwischen  $\epsilon$ -PPM und SMS-EMOA bzw. zwischen  $\epsilon$ -PPM und NSGA-2 keine signifikanten Qualitätsunterschiede bezüglich des Hypervolumens festzustellen, während ZDT2 am besten vom  $\epsilon$ -PPM approximiert wird. Erstaunlich ist hier, dass der entwickelte Mechanismus die Front schneller finden kann als dies mit NSGA-2 und SMS-EMOA möglich ist. Beide Vergleichsalgorithmen versagen bei der Anzahl von 10.000 Funktionsauswertungen oftmals, so dass keine Lösung im Zielbereich gefunden wird. Dies erklärt insbesondere die großen Schwankungen in den Ergebnissen von SMS-EMOA. Zusätzliche, hier nicht weiter dokumentierte

Experimente haben jedoch gezeigt, dass dieses Problem mit einer größeren Anzahl von Funktionsauswertungen verschwindet.<sup>4</sup> Die starke Fokussierung der Räuber auf ein einzelnes Kriterium trägt zwar, wie bereits diskutiert, zum Diversitätsverlust bei, unterstützt durch die Tendenz in global optimale Extremlösungen jedoch auch das schnelle Auffinden des eigentlichen Zielgebietes der Optimierung. Wird die Extremwerttendenz anschließend, wie vorgeschlagen, durch einen geeigneten Mechanismus kompensiert, kann dies zu einer guten Approximation der Lösungsmenge führen.

## 8.4 Diskussion der Erweiterung

In diesem Kapitel wurde der Grundgedanke des  $\epsilon$ -Constraint-Ansatzes – nämlich ein einziges Kriterium zu verfolgen und alle übrigen Kriterien durch obere Schranken zu ersetzen – in das Räuber-Beute-Modell integriert, um unterschiedliche Gebiete der Paretofront zu fokussieren. So erreichen wir mittelfristig eine vollständige Abdeckung der Paretofront. Die zusätzlich eingeführte Kooperationsstrategie für die Räuber nutzt die bekannte Tendenz der Lösungen in Extembereiche der Paretofront, und trägt zur gemeinsamen Erkundung des angestrebten Zielgebietes bei.

Wir konnten zeigen, dass dieses zweistufige Verfahren insbesondere in hochdimensionalen Entscheidungsräumen, etwa der ZDT1-3-Probleme, konkurrenzfähig zu etablierten Ansätzen ist und ebenfalls nicht-konvexe Probleme behandeln kann. Dies konnte zudem ohne nebenläufige lokale Suchstrategien, durch die ausschließliche Betrachtung des Bildraums, erreicht werden. Es ist aber auch darauf hinzuweisen, dass die Approximationsqualität des  $\epsilon$ -PPM für die Probleme ZDT4 und ZDT6 mit 10.000 Funktionsauswertungen nicht hinreichend ist. Die Vergleichsmethoden NSGA-II und SMS-EMOA liefern dort bessere Ergebnisse. Der Räuber-Beute-Ansatz stagniert oftmals in einer lokalen Paretofront. Dies ist darin begründet, dass die einkriterielle Suche das globale Optimum nicht erreicht. Im Kontext dieses Ansatzes – der zuerst die globalen Optima aufsucht und anschließend Kompromisse generiert – kann erneut über den Einsatz von Adaptationsverfahren für die Mutationsschrittweite nachgedacht werden, um die Approximation des Zielbereiches sicherzustellen.

Insgesamt konnten wir in diesem Kapitel erneut nachweisen, dass es durch die modulare Struktur des Modells durchaus möglich ist, auf direktem Wege Modifikationen am Modell vorzunehmen, die analysierten und verstandenen Eigenschaften zu nutzen und zugleich die Dynamik beherrschbar zu erweitern. Diese Erkenntnis motiviert im abschließenden Teil der Arbeit auch unsere Übertragung des Räuber-Beute-Modells in den Bereich der ganzzahligen Optimierung. Am Beispiel des Maschinenschedulings versuchen wir, die praktische Anwendbarkeit der Methode zu

---

<sup>4</sup>Eine kompetitive Untersuchung von Algorithmen ist nicht Gegenstand dieser Evaluation. Vielmehr steht die Bewertung der Erweiterung des Räuber-Beute-Modells als algorithmisches Prinzip im Vordergrund.

beurteilen und zugleich den Aspekt der Modularität für die Integration von Expertenwissen zu nutzen.

## Teil IV

# Anwendung in der diskreten Optimierung - Eine Fallstudie im mehrkriteriellen Scheduling



## Kapitel 9

# Mehrkriterielles Scheduling – Problemkontext und Grundlagen

Scheduling – gemeint ist die Zuteilung knapper Ressourcen zu gegebenen Aufgaben über die Zeit – beschreibt eine Klasse von kombinatorischen Optimierungsproblemen, die in der heutigen Zeit, geprägt von der Erbringung von Dienstleistungen, eine immer stärkere praktische Bedeutung erlangen. Logistik, Transport und Verkehr, die Produktionsplanung, das Personalmanagement und die Informationstechnologie sind nur einige Bereiche, in denen diese Optimierungsprobleme eine zentrale Rolle spielen. Diese Bereiche sind so zugleich Beispiele für das breite Anwendungsfeld potentieller Lösungsverfahren. Durch den direkten Praxisbezug vieler Schedulingprobleme hat sich in diesem Forschungsbereich eine Theorie herausgebildet, die neben einer sehr praxisnahen Notation auch anwendungsnahe Modelle zur theoretischen Betrachtung der Schedulingprobleme verwendet.

Im Verlaufe des 20. Jahrhunderts ist es im Forschungsgebiet des Scheduling gelungen, eine große Menge an theoretischen Untersuchungen und Ergebnissen, aber auch Heuristiken und „Daumenregeln“ für die Behandlung von *einkriteriellen Schedulingproblemen* zur Verfügung zu stellen. Wie für viele praktische Problemstellungen im Bereich der Optimierung gilt jedoch auch beim Scheduling, dass bei realen Schedulingproblemen nicht ein einziges, sondern oft mehrere Kriterien zugleich optimiert werden sollen. Die Betrachtung solcher Probleme, speziell im Scheduling, begann jedoch tatsächlich erst in den achtziger Jahren des vorherigen Jahrhunderts (Hoogeveen, 2005). So finden sich in der Literatur nur wenige deterministische Ansätze für das *mehrkriterielle Scheduling*. Für die existierenden Verfahren gilt, dass sie entweder ausschließlich einfache Modelle betrachten oder auf generelle und iterative Lösungsansätze, wie in Kapitel 3 vorgestellt, setzen. Daneben existiert natürlich eine gewisse Menge an randomisierten Ansätzen, die oft jedoch sehr spezialisiert sind oder sich ausschließlich auf Zufall und Richtungsselektion im Sinne genereller evolutionärer Heuristiken verlassen. Generelle Ansätze

zur Integration von einkriteriellem Wissen und mehrkriteriellen Verfahren sind in diesem Bereich spärlich zu finden.

Dieses Kapitel soll, als Vorbereitung auf die Untersuchung des Räuber-Beute-Modells im Kontext von Schedulingproblemen, eine knappe Einführung in das Thema geben und einige wichtige Ergebnisse für die spätere Verwendung einführen. Dazu definieren wir zuerst Schedulingprobleme und die übliche Notation. Danach betrachten wir einkriterielle Problemstellungen und zugehörige Lösungsstrategien. Bevor wir die von uns näher untersuchten mehrkriteriellen Schedulingprobleme definieren, werden noch ein einfaches Referenzproblem und der zugehörige Lösungsansatz aus dem mehrkriteriellen Scheduling betrachtet.

## 9.1 Scheduling und Notation

Der Begriff des *Scheduling* beschreibt das Problem der Zuweisung von wenigen Ressourcen  $\mathbb{M} = \{M_1, \dots, M_k\}$ , oftmals auch als Maschinen bezeichnet, zu  $n$  Jobs  $J = \{J_1, \dots, J_n\}$  über die Zeit. Dabei gilt  $n, k \in \mathbb{N}$  und zumeist  $k < n$ , da sonst kein Schedulingproblem existiert. Jeder Job hat einige wichtige Eigenschaften, die ihn beschreiben, siehe auch Abbildung 9.1: den Einreichungszeitpunkt (engl. *release date*)  $r_j$  und damit frühesten Startzeitpunkt des Jobs, die Bearbeitungszeit (engl. *processing time*)  $p_j$ , den Fertigstellungszeitpunkt (engl. *completion time*)  $C_j$  und das Fälligkeitsdatum (engl. *due date*)  $d_j$ . Pinedo (2008) beschreibt alle Eigenschaften sehr ausführlich und nennt weitere Merkmale wie die Gewichtung bzw. Priorität  $w_j$  des Jobs  $j$  oder die Deadline  $\bar{d}_j$  als strikte Variante des Fälligkeitsdatums, welche wir in unseren Problemstellungen jedoch nicht verwenden.

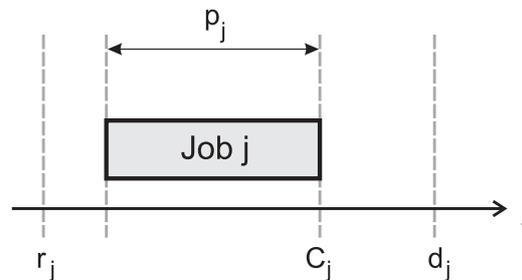


ABBILDUNG 9.1: Schematische Darstellung der Job-Eigenschaften. Dabei beschreibt  $p_j$  die Bearbeitungszeit,  $r_j$  den Einreichungszeitpunkt,  $C_j$  den Fertigstellungszeitpunkt und  $d_j$  das Fälligkeitsdatum des Jobs  $j$ .

Abgesehen von den Job-Eigenschaften werden Maschinenumgebungen in Klassen eingeteilt. Die einfachste Klasse stellen die Einmaschinenmodelle dar, die selbst ein Spezialfall des Modells identischer und paralleler Maschinen sind. Während letztere mit  $P_k$ , bezeichnet<sup>1</sup> werden ( $k$  entspricht der Anzahl der Maschinen), wird

<sup>1</sup>In der Literatur wird dieses Modell eigentlich mit  $P_m$  bezeichnet. Um in dieser Arbeit jedoch einen Konflikt mit  $m$  als Anzahl der Kriterien zu vermeiden, verwenden wir hier alternativ die Notation  $P_k$ .

für das Einmaschinenmodell die besondere Bezeichnung 1 eingeführt. Auch hier erläutert Pinedo (2008) weitere Modelle, die wir im Detail nicht betrachten.

Die Formulierung der angestrebten Ziele beim Scheduling<sup>2</sup> geschieht zumeist über die Gesamtheit der Jobs und ihrer Eigenschaften. So beschreibt  $C_{max} = \max_{j=1, \dots, n} \{C_j\}$  die Produktionsspanne (engl. *makespan*), also die späteste Fertigstellung eines Jobs. Die Betrachtung  $\sum C_j = \sum_{j=1}^n C_j$  entspricht der Summe aller Fertigstellungszeiten (engl. *total completion time*) und gibt an, wie effizient ein System die Gesamtmenge der Jobs bearbeitet.<sup>3</sup> Bezüglich des Fälligkeitsdatums können ebenfalls Ziele formuliert werden. So bezeichnet  $L_j = C_j - d_j$  die Verspätung (engl. *lateness*) eines Jobs  $j$  und  $L_{max} = \max_{j=1, \dots, n} \{L_j\}$  die maximale Verspätung (engl. *maximum lateness*). Beide Zielfunktionen können durchaus negative Werte annehmen. Die Verzögerung (engl. *tardiness*) nimmt entsprechend  $T_j = \max(L_j, 0)$  nur für tatsächlich zu spät fertiggestellte Jobs einen von 0 unterschiedlichen Wert an, und wird im Rahmen der sogenannten Gesamtverzögerung (engl. *total tardiness*)  $\sum T_j = \sum_{j=1}^n T_j$  aggregiert. Schließlich ermittelt  $\sum U_j = \sum_{j=1}^n U_j$  mit  $U_j = 1$  für  $L_j > 0$  und andernfalls  $U_j = 0$ , die Anzahl der verspäteten Jobs.

Im Folgenden nutzen wir die von Graham, Lawer, Lenstra & Rinnooy Kan (1979) entwickelte Drei-Felder-Notation  $\alpha|\beta|\gamma$ . Darin spezifiziert  $\alpha$  die Maschinenumgebung. Weiterhin enthält  $\beta$  Restriktionen bzw. Randbedingungen, wenn diese vorhanden sind. Andernfalls bleibt dieses Feld leer. Schließlich werden im  $\gamma$ -Feld die Ziele der Optimierung eingetragen. Wir verzichten an dieser Stelle auf Beispiele, da verschiedene Probleme bereits in den nächsten Abschnitten eingeführt werden. Für unsere Probleme mit  $m$  Kriterien verwenden wir im  $\gamma$ -Feld die von T'kindt & Billaut (2006) durch  $\#(\gamma_1, \dots, \gamma_m)$  dargestellte gleichberechtigte Aufzählung der Ziele und schreiben in der Drei-Felder-Notation verkürzt  $\alpha|\beta|\gamma_1, \dots, \gamma_m$ , wenn keine Verwechslungsgefahr besteht.

## 9.2 Einkriterielle Schedulingprobleme

Wir stellen nun einige grundlegende Schedulingprobleme vor, die entweder optimal lösbar sind, durch Regeln mit einer garantierten Güte gelöst werden können oder für die es eine gute Heuristik gibt. Diese Regeln und Strategien nehmen wir später als Ausgangspunkt für die Lösung mehrkriterieller Probleme mit Hilfe des Räuber-Beute-Modells. Insgesamt betrachten wir nur einen sehr kleinen Ausschnitt existierender Schedulingprobleme. Eine gute Darstellung vieler anderer Probleme findet sich bei Pinedo (2008).

Im Kontext der Einmaschinen-Umgebung sind einige Probleme effizient lösbar. Das Problem  $1||C_{max}$  ist sicherlich das einfachste Problem, da jeder Schedule ohne Verzögerung optimal ist. Für das Problem  $1||\sum C_j$  ist offensichtlich, dass eine

<sup>2</sup>Wir betrachten auch hier immer Minimierungsprobleme.

<sup>3</sup>In einem Fertigungsprozess kann diese Systemmetrik etwa eine Information über entstehende Lagerkosten für Materialien zu einer festen Auftragsmenge (Jobs) liefern (Pinedo, 2008).

aufsteigende Sortierung der Jobs nach Bearbeitungszeiten optimal ist. Diese Regel wird in der Literatur als SPT (shortest processing time first) bezeichnet (Pinedo, 2008). Das Problem ist ein Spezialfall von  $1||\sum w_j C_j$ , welches mit der sogenannten WSPT-Regel (weighted shortest processing time) gelöst wird, indem die Jobs aufsteigend entsprechend des Verhältnisses  $\frac{w_j}{p_j}$  angeordnet werden (Smith, 1956). Unter der Berücksichtigung von Einreichungszeitpunkten wird dieses Problem NP-schwer (Lenstra, Kan & Brucker, 1977). Für das Problem  $1||L_{max}$  ergibt sich mit der EDD-Regel (earliest due date first) eine optimale aufsteigende Sortierung nach Fälligkeitsdaten der Jobs (Jackson, 1955). Erweitert um Einreichungszeitpunkte, wird auch das Problem  $1|r_j|L_{max}$  NP-schwer. Das Problem  $1||\sum U_j$  kann mit dem Algorithmus von Moore (1968) gelöst werden, der die Jobs zuerst nach aufsteigenden Fälligkeitsdaten auf die Maschinen bringt und beim Auftreten einer Verspätung den Job mit größter Bearbeitungszeit aus der Sequenz entfernt und als verspätet markiert. Fügt man Einreichungsdaten als Bedingung hinzu, so wird das Problem NP-schwer.

Betrachtet man die Umgebung der parallelen identischen Maschinen, so steigt die Komplexität der Schedulingprobleme meist an. Das Problem  $P_k||C_{max}$  ist NP-schwer, kann aber durch eine absteigende Sortierung bezüglich der Bearbeitungszeiten (LPT - longest processing time first) mit einer Qualitätsgarantie von  $C_{max}(LPT) \leq (\frac{4}{3} - \frac{1}{3m}) \cdot C_{max}(OPT)$  gelöst werden. Das verwandte Problem  $P_k||L_{max}$  ist ebenfalls NP-schwer. Hingegen bleibt das Problem  $P_k||\sum C_j$  einfach und ist auch weiterhin mit SPT lösbar. Für die allgemeinere Variante  $P_k||\sum w_j C_j$  haben Kawaguchi & Kyan (1986) sowie kürzlich Schwiegelshohn (2011) in einer kürzeren Version gezeigt, dass WSPT einen Approximationsfaktor von  $\frac{1}{2}(1 + \sqrt{2})$  besitzt. Das Problem  $P_k||\sum U_j$  ist NP-schwer (Garey & Johnson, 1979). Trotzdem findet sich bei Süer, Báez & Czajkiewicz (1993) eine gute Heuristik, die das Problem nach allen bisherigen empirischen Untersuchungen oftmals sehr exakt löst. Eine Qualitätsgarantie wurde für den Ansatz bisher nicht ermittelt. Die Autoren nehmen sich dabei den zuvor erwähnten Algorithmus von Moore zum Vorbild und planen die Jobs sukzessive nach aufsteigenden Fälligkeitsdaten jeweils auf der Maschine mit geringster Produktionsspanne. Wird für den Job eine Verspätung festgestellt, so wird von der betreffenden Maschine der größte Job entfernt. Dieser wird als verspätet markiert. Wir werden diese Heuristik im Folgenden als SBC3 bezeichnen. Die Autoren schlagen noch zwei weitere Varianten vor, die jedoch kaum von der zuvor beschriebenen abweichen und daher hier nicht weiter betrachtet werden.

Als wichtiges Ergebnis dieser kurzen Betrachtung können wir festhalten, dass es für grundlegende Probleme im Bereich des einkriteriellen Scheduling Algorithmen oder Regeln gibt, die eine optimale oder zumindest sehr gute Sequenzierung von Jobs sicherstellen. Haupt (1989) fasst viele davon in seiner Arbeit zusammen. Hier sind diejenigen, die sich aus der vorherigen Darstellung ergeben in Tabelle 9.1 zusammengefasst.

TABELLE 9.1: Zusammenfassung der wichtigsten Regeln zur Lösung grundlegender Schedulingprobleme. Wir beschränken uns auf die in dieser Arbeit diskutierten Probleme. Optimale Regeln sind mit einem ✓ markiert.

Schedulingproblem	Regel	Opt.
<i>Einzelmaschinen</i>		
$1  C_{max}$	Jede Sequenz ohne Verzögerung	✓
$1  \sum C_j$	SPT (shortest processing time first)	✓
$1  \sum w_j C_j$	WSPT (weighted SPT)	✓
$1  L_{max}$	EDD (earliest due date first)	✓
$1  \sum U_j$	Moore-Algorithmus	✓
<i>Parallele identische Maschinen</i>		
$P_k  C_{max}$	LPT (longest processing time first)	
$P_k  \sum C_j$	SPT	✓
$P_k  \sum w_j C_j$	WSPT	
$P_k  \sum U_j$	SBC3-Algorithmus	

### 9.3 Mehrkriterielle Schedulingprobleme

Die Betrachtung mehrerer Kriterien gleichzeitig ist auch im Bereich des Scheduling eine natürliche Entwicklung. Hoogeveen (2005) stellt dar, dass seit den späten 80er Jahren diese Betrachtung begann und sich daraus ein eigenes Forschungsgebiet entwickelt hat. Umso erstaunlicher erscheint auf den ersten Blick die Feststellung, dass sich die Ergebnisse dieser Forschung auch fast 30 Jahre später fast ausschließlich auf theoretische Resultate zur Komplexität von Problemen beschränken. Dies liegt aber sicher an der Natur der Probleme. Betrachtet man mehrere Kriterien, so sind fast alle Problemstellungen NP-schwer (Chen & Bulfin, 1993). Es gibt fast keine Probleme, die in polynomieller Zeit optimal gelöst werden können, noch sind Regeln für die Sequenzierung von Jobs zu finden. Dies ist wiederum nicht weiter überraschend, da eine Sequenzierung nur eine einzige Lösung erzeugt, in der mehrkriteriellen Optimierung aber grundsätzlich eine Lösungsmenge von Kompromissen angestrebt wird. Oftmals können nur die bereits in Kapitel 3 dargestellten iterativen Verfahren genutzt werden, um eine Lösung durch Angabe von Gewichtungen, Zenitpunkten oder durch Verwendung der  $\epsilon$ -Constraint-Methode zu erreichen. Einen Einstieg vermittelt die Arbeit von Dutot, Rzdca, Saule & Trystram (2010).

Eines der wenigen in polynomieller Zeit optimal lösbaren Probleme findet sich in der Einmaschinenumgebung und betrachtet die Gesamtbearbeitungszeit und die maximale Verspätung als Kriterien:  $1||\sum C_j, L_{max}$ . Das Problem und der von van Wassenhove & Gelders (1980) zur optimalen Lösung vorgeschlagene Algorithmus sollen hier aus zwei Gründen betrachtet werden. Einerseits bietet dieses Problem durch die Verfügbarkeit einer optimalen Lösung einen guten Einstiegspunkt für die Erprobung des Räuber-Beute-Modells im Kontext des Scheduling. Andererseits zeigt der Algorithmus anschaulich, wie existierende Regeln für die einkrite-

rielle Optimierung zu einem Ansatz der mehrkriteriellen Optimierung zusammengefügt werden können. Über die Regeln EDD und SPT/EDD<sup>4</sup> ist es möglich, die beiden Extremlösungen der Paretofront aufzufinden. Ausgehend von der Lösung  $L_{max}(EDD)$ , wird die beste Lösung für das Ziel  $\sum C_j$  und das minimale Inkrement  $\ell$  in  $L_{max}$  bestimmt, unter dem eine Verbesserung für  $\sum C_j$  erreicht werden kann. Das ursprüngliche Problem wird durch  $d_j + \ell$  in ein neues Problem transformiert. Danach wird erneut versucht, die Sequenz bzgl.  $\sum C_j$  unter der Bedingung  $L_{max} = \ell$  optimal zu planen und ein neues minimales Inkrement  $\ell'$  bestimmt. Dieser Prozess wird fortgesetzt, bis der zuvor ermittelte Wert von  $L_{max}(SPT/EDD)$  erreicht wird. Alle ermittelten Zwischenergebnisse definieren die Paretofront. Die Laufzeit des Algorithmus kann durch  $\mathcal{O}(n^3 \log n)$  abgeschätzt werden.

Als Beispiele für NP-schwere mehrkriterielle Schedulingprobleme sollen in dieser Arbeit mehrere Probleme unterschiedlichen Schwierigkeitsgrades betrachtet werden. Ein sehr anschauliches Beispiel für gegensätzliche Kriterien und noch gegensätzlichere einkriterielle Regeln, ist sicherlich das oft als MAXANDSUM abgekürzte Problem  $P_k || C_{max}, \sum C_j$  (Dutot et al., 2010). Obwohl  $P_k || \sum C_j$  mit SPT einfach zu lösen ist, wird das gesamte Problem durch die Komplexität von  $P_k || C_{max}$  NP-schwer (Chen & Bulfin, 1993). Auch die Zielsetzung ist unterschiedlich. Strebt das Kriterium  $C_{max}$  einen sehr balancierten Schedule an, so kann dies für die Gesamtbearbeitungszeit durchaus nachteilig sein. Fast plakativ wird der Unterschied jedoch, wenn man die optimalen und heuristischen Regeln für beide Probleme separat betrachtet: Die für  $\sum C_j$  optimale Sortierung geschieht aufsteigend für die Bearbeitungszeit, während die beste Regel für  $C_{max}$  mit LPT direkt gegensätzlich ist. Zwar garantiert SPT auch für  $C_{max}$  einen Approximationsfaktor von  $2 - \frac{1}{m}$ , dieser ist jedoch schlechter als die durch LPT erreichbare Lösung.<sup>5</sup> Ein weiteres Beispiel ist etwa die Betrachtung von  $P_k || \sum C_j, \sum U_j$ , oder schwerer von  $P_k | r_j | \sum C_j, \sum U_j$ , wobei die Einreichungszeitpunkte dafür sorgen, dass selbst jedes einkriterielle Teilproblem NP-schwer wird. Die Kombination aller drei Kriterien  $P_k || C_{max}, \sum C_j, \sum U_j$  ist dabei bisher nur von Grimme, Lepping & Schwiigelshohn (2011) behandelt worden. Diese Arbeit werden wir im folgenden Kapitel näher betrachten.

<sup>4</sup>Nachdem die Sequenz bzgl. SPT sortiert wurde, werden Jobs mit gleicher Bearbeitungszeit nach Fälligkeitsdatum sortiert.

<sup>5</sup>Grundsätzlich gilt, dass eine beliebige Sortierung (LIST) für  $P_k || C_{max}$  den angegebenen Approximationsfaktor garantiert, siehe auch Pinedo (2008).

## Kapitel 10

# Untersuchung der Anwendung des Räuber-Beute-Modells auf mehrkriterielle Ein- und Mehrmaschinenprobleme

Die knappe Diskussion im vorhergehenden Kapitel hat gezeigt, dass es eine Reihe von guten Verfahren und Regeln zur Lösung einkriterieller Probleme gibt. Im gleichen Maße sind solche Regeln für mehrkriterielle Probleme jedoch nicht zu finden. Die Theorie ermöglicht zwar grundlegende Aussagen bezüglich der Problemkomplexität, ein Transfer der Methoden und Heuristiken aus dem einkriteriellen Bereich in die mehrkriterielle algorithmische Anwendung ist aber schwierig. Grundsätzlich ist nicht zu erwarten, dass eine mehrkriterielle Lösung mittels einer einfachen Regel bestimmt werden kann. Es ist oftmals aber auch nicht ersichtlich, wie einzelne für einkriterielle Probleme gut funktionierende Regeln oder Ansätze sinnvoll kombiniert werden können, um eine mehrkriterielle Lösung zu erzeugen. Dass dies wie im zuvor diskutierten Problem  $1||L_{max}, \sum C_j$  durch die Kombination von EDD und SPT funktioniert, ist eher ungewöhnlich.

Die allgemeinen und oftmals sehr monolithischen heuristischen Verfahren der mehrkriteriellen Optimierung – sowohl deterministische, wie auch natur-inspirierte – bieten hier keine Hilfestellung. Mit anderen Worten, es ist nicht möglich, die Ergebnisse der einkriteriellen Optimierung „einfach“ in einem Algorithmus zu aggregieren. Es müssen vielmehr ganz spezielle Verfahren für ein gegebenes Problem entwickelt oder alternative algorithmische Ansätze gewählt werden, die die Integration bereits existierenden einkriteriellen Wissens erlauben. Grundsätzlich sollte bei einem solchen Verfahren auch beachtet werden, dass die Integration von einkriteriellem Wissen für ein spezielles Problem nicht jeweils mit einem weitgehenden Neuentwurf der algorithmischen Struktur oder zumindest großen Änderungen einhergeht. Die modulare Struktur und die nebenläufigen einkriteriellen Suchen im Räuber-Beute-Modell deuten darauf hin, dass der Ansatz sich gerade für die Um-

setzung dieser Anforderungen einsetzen lässt.

Somit ergibt sich eine zweiteilige Zielsetzung für die Betrachtung des Modells in diesem Kapitel:

1. Zuerst betrachten wir den Transfer des Räuber-Beute-Modells in die Domäne des Scheduling und überprüfen seine Einsetzbarkeit grundlegend. Dazu müssen wir die Repräsentation der Beute anpassen und damit auch die Variationsoperatoren. Neben der Verarbeitung eines anderen Genotyps müssen diese auch in der Lage sein, heuristisches Wissen in den Evolutionsprozess einzubringen.
2. Danach können wir die Erweiterbarkeit des modularen Ansatzes bezüglich neuer Problemstellungen und bezüglich der Hinzunahme neuer Kriterien überprüfen. Dies erlaubt schließlich eine erste Aussage über die praktische Anwendbarkeit des Ansatzes als modulares und allgemeines Lösungsverfahren im Bereich des Scheduling.

Es sei bereits jetzt erwähnt, dass viele der hier vorgestellten Ergebnisse auf den Arbeiten von Grimme, Lepping, Papaspyrou, Kemmerling und Schwiiegelshohn (Grimme, Lepping & Papaspyrou, 2008b; Grimme, Kemmerling & Lepping, 2011; Grimme et al., 2011; Grimme & Lepping, 2011a) basieren.

## 10.1 Transfer des Räuber-Beute-Modells auf Probleme im Scheduling

Abweichend von der bisherigen Anwendung des Räuber-Beute-Modells auf reellwertige Probleme, macht die Nutzung des Ansatzes auf ganzzahligen Schedulingproblemen eine Veränderung der Repräsentation der Lösung, also der Beute, notwendig. Dadurch bedingt müssen auch die Variationsoperatoren angepasst werden. Da wir anstreben, die verfügbaren Regeln des einkriteriellen Scheduling aufzunehmen, entwickeln wir einen speziellen Operator, der dies erlaubt. Schließlich gilt es noch die Kopplung der Räuber und der neu entwickelten Operatoren betrachten.

### 10.1.1 Repräsentation von Schedulingproblemen

Da wir in unserem Problemkontext nur Schedulinginstanzen mit fester Jobanzahl  $n$  nutzen, können wir für die genotypische Repräsentation eines Schedules eine einfache Permutationsdarstellung der Jobnummern verwenden. Somit wird jedes Beuteindividuum durch einen  $n$ -dimensionalen Vektor repräsentiert. Da wir hier grundsätzlich sogenannte Offline-Probleme betrachten, also solche, in denen alle Informationen über die Jobs zu Beginn der Betrachtung verfügbar sind, ist das Ziel der Optimierung, eine optimale Permutation der Jobs aufzufinden. Die Überführung der Sequenz in einen auswertbaren Schedule und Phänotypen geschieht für Ein- und Mehrmaschinenumgebungen daher durch Anwendung einer

einfachen First Come First Serve (FCFS)-Regel. Dies bedeutet, dass wir bei der Konstruktion des Schedules mit der gegebenen Sequenz der Lösung und leeren Maschinen starten. Dann werden die Jobs entsprechend der Vorgabe der Sequenz auf die Maschinen gebracht. Immer wenn eine Maschine die Bearbeitung eines Jobs abgeschlossen hat, kann der nächste Job in der Sequenz bearbeitet werden. Abbildung 10.1 zeigt an einem einfachen Beispiel mit acht Jobs und zwei Maschinen, wie eine Sequenz in einen Schedule überführt wird. Aus dem entstandenen Schedule (hier rechts als Gantt-Chart dargestellt), lassen sich schließlich die Werte für die Kriterien ableiten.

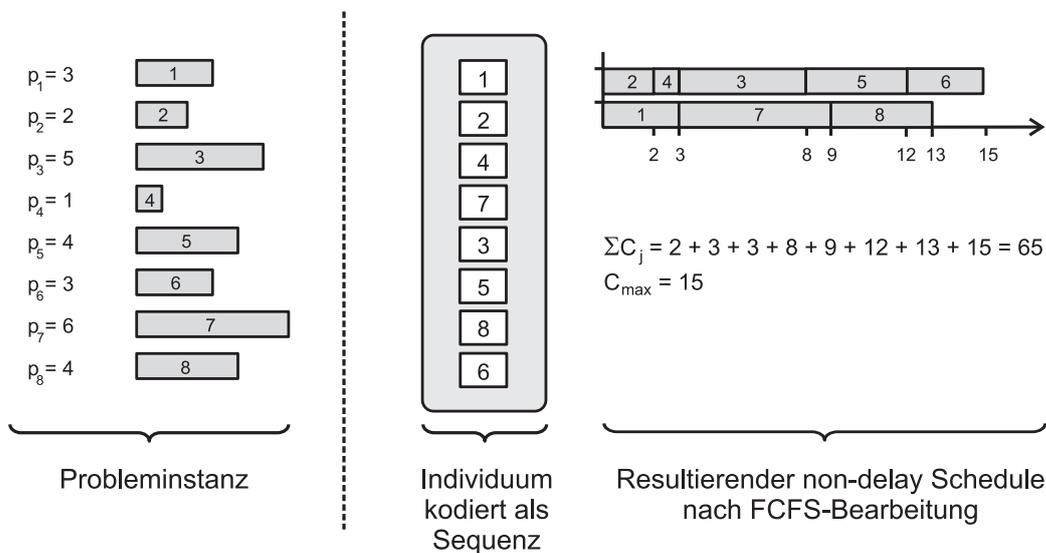


ABBILDUNG 10.1: Darstellung des Schedules als Permutation auf der genotypischen Ebene und schematische Beschreibung der Überführung der Sequenz in den Schedule, die phänotypische Darstellung. Für Ein- und Mehrmaschinenprobleme geschieht dies durch First-Come-First-Serve (FCFS)-Einreichung der Jobs, nach Grimme et al. (2011).

### 10.1.2 Ein Variationsoperator für Schedulinganwendungen

Im Falle der einkriteriellen evolutionären Algorithmen ist es üblich, spezielles Problemwissen direkt durch die Variationsoperatoren einfließen zu lassen. Beispiele für das Scheduling in Ein- und Mehrmaschinenumgebungen, aber auch für Stundenplanprobleme finden sich etwa bei Lee (1994), Franca, Mendes & Moscato (2001), Cheng & Gen (1996), sowie bei Burke, Newall & Weare (1996, 1998). Eine sehr umfassende Darstellung dieses Teilgebiets der sogenannten memetischen Algorithmen – also jener Ansätze, die lokale Lern- oder Suchverfahren in globale Suchschemata einbringen – findet sich bei Moscato, Cotta & Mendes (2004). Im mehrkriteriellen Bereich finden sich annähernd keine Ansätze der Integration von Wissen über die Variationsoperatoren. Dies mag darin begründet sein, dass in den meisten Algorithmen die Selektion und nicht die Variation als Forschungsgegenstand betrachtet

wird. Stärker wiegt aber sicherlich das Problem, verschiedene und oft gegensätzliche Regeln und Heuristiken in einer einzigen lokalen Suche oder einem einzigen Operator zusammenzufassen. Dies ist dann nur eine Verschiebung des eigentlichen mehrkriteriellen Problems auf die Ebene der Variationsoperatoren.

Da wir im Räuber-Beute-Modell nur einkriteriell selektieren und die Operatoren direkt an die Räuber binden, die erst in ihrem Zusammenwirken eine mehrkriterielle Lösung erzeugen, beschränken wir uns auch auf den Einsatz einkriterieller Variationsoperatoren. Es reicht also aus, das jeweilige Wissen als Operatormodul darzustellen. Für diese Integration von Wissen entwickeln wir hier einen generellen Operator: Mit ihm ist es möglich, die Umordnung von Sequenzen (z.B. motiviert aus SPT, LPT oder EDD) als zufälligen Einfluss umzusetzen. Im Grunde entsteht so eine Mutation, die im Sinne der entsprechenden (Sortierungs-)Regel eine Tendenz aufweist.

Die Grundidee des Operators ist es, die Sortierungen, die durch Regeln vorgegeben werden, gut dosiert aber zufällig in den Genotyp einfließen zu lassen. Dazu wird eine Schrittweite  $\sigma$  festgelegt und für ein Individuum eine zufällige Position im Genom gewählt. Auf Grundlage der festgelegten Mutationsschrittweite  $\delta$  wird ein ganzzahliger positiver Zufallswert  $\mathbb{N} \ni z \sim |\mathcal{N}(0, \sigma)|$  bestimmt. Zu beiden Seiten der Position im Genom wird eine Sequenz der maximalen Länge  $z$  und damit eine Gesamtsequenz mit maximaler Länge von  $2z + 1$  Genen ausgewählt. Die rechten und linken Grenzen des Genoms beschränken die Auswahl. Diese Teilsequenz wird entsprechend der durch die Regel vorgegebenen Sortierung angeordnet. Durch eine Änderung der Schrittweite  $\sigma$  ist es also möglich, die Größe der betrachteten Teilsequenz und zugleich die Auswirkung der Sortierung zu beeinflussen. Als Extremeinstellungen können  $\sigma = 0$  und  $\sigma = n$  gewählt werden. Die erste Einstellung schaltet den Einfluss des Operators aus. Die zweite Einstellung führt in den meisten Fällen zu einer Anwendung der Sortierregel auf das gesamte Genom. Abbildung 10.2 zeigt die Anwendung für ein  $z = 2$  und eine Sortierung der fünf Gene enthaltenden Teilsequenz gemäß SPT. Bezüglich des Kriteriums  $\sum C_j$  trägt diese Umordnung zur Verbesserung bei.

### 10.1.3 Die Kopplung von Räubern und Variationsoperatoren

Um den Transfer des Räuber-Beute-Modells auf Schedulingprobleme abzuschließen, muss noch die Verwendung der Variationsoperatoren in Verbindung mit den Räubern diskutiert werden. Bei der Betrachtung reellwertiger Probleme und der Verwendung von Variationsoperatoren ohne Richtungstendenz stellte sich diese Frage nur bezüglich der systemweiten Kombination von Einflüssen. Mit dem vorgeschlagenen Operator zur Integration von Schedulingregeln und damit einer Tendenz der Einflüsse, ergibt sich eine neue Unterteilung in der Kopplung von Selektionsfaktor und Variationsoperator: Einerseits kann ein Räuber mit einem für sein Kriterium zuträglichen Operator verbunden werden, d.h. ein Räuber der bezüglich  $\sum C_j$  selektiert, kann mit einem den SPT-Einfluss vertretenden Operator ausge-

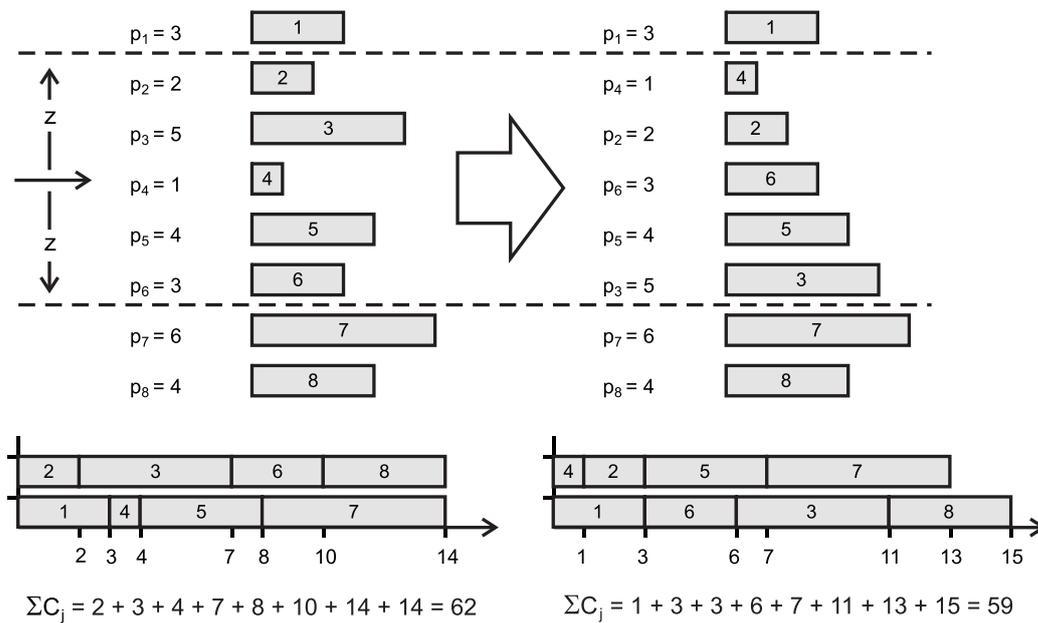


ABBILDUNG 10.2: Schematische Darstellung der Funktionsweise des entwickelten Variationsoperators zur Integration von auf Sortierungen basierten heuristischen Regeln aus dem Gebiet des einkriteriellen Scheduling, nach Grimme et al. (2011).

stattet werden. Andererseits ist es aber auch möglich, einen Räuber mit einem für ein anderes Kriterium vorteilhaften Operator auszustatten.

Im ersten Falle wird die Anwendung eines dem verfolgten Kriterium entsprechenden Operators zu einer Verstärkung des vom Räuber eingebrachten Kriteriums führen. Zusätzlich zur Selektion bezüglich des Kriteriums wird die Beute vom mitgeführten Operator auf heuristischem Weg stark in Richtung des Kriteriums gedrängt. Durch eine solche Kopplung können wir eine Verstärkung des Räubereinflusses und damit eine Konvergenz in die Extremwerte der Paretofront unterstützen.

Im zweiten Falle arbeitet der Operator gegen den Einfluss des vom Räuber verfolgten Kriteriums und führt zu einem lexikographischen Ansatz in der Optimierung. So ist das Räuber-kriterium das primäre Ziel und konserviert die Güte der Beute bezüglich dieses Ziels; der Variationsoperator verfolgt aber in diesem Rahmen, das von ihm unterstützte Ziel quasi nachgeordnet. Dies kann zur Diversitätserhaltung innerhalb der Population beitragen und ist daher eine Kopplungsvariante, die in Kombination mit der zuvor beschriebenen Verstärkung eingesetzt werden kann. Beide zusammen können zu einer gut abgedeckten Paretofront führen.

Hier liegt auch der Unterschied zu der vorherigen Betrachtung reellwertiger Probleme. Wir werden in diesem Kapitel ausschließlich die Einsetzbarkeit des Räuber-Beute-Modells unter dem Einfluss des entwickelten Variationsoperators betrachten. Um die Betrachtung nicht zu verzerren, setzen wir keine zusätzliche Rekombination ein. Ausschließlich ungerichtete Mutation im Sinne von zufälligen

Positionsvertauschungen (Swap-Mutation) innerhalb des Genoms wird eingesetzt, um einer vollständigen Sortierung einzelner Genome und damit einem Verlust an Diversität entgegen zu wirken. An entsprechender Stelle wird auf die Konfiguration jeweils eingegangen.

## 10.2 Anwendung auf ein Einmaschinenproblem

Die erste Untersuchung zur Übertragung des Räuber-Beute-Modells wollen wir auf das Einmaschinenproblem  $1||\sum C_j, L_{max}$  beschränken (Grimme et al., 2008b, 2011). Für dieses Problem können wir mit Hilfe des in Abschnitt 9.1 dargestellten Algorithmus von van Wassenhove & Gelders (1980) eine optimale Referenzlösung bestimmen und direkt feststellen, ob die zuvor beschriebene Funktionsweise des vorgeschlagenen Variationsoperators und die diskutierten Eigenschaften der Kombination von Selektion und Variation nachgewiesen werden können. Zur Untersuchung des Problems generieren wir eine zufällige Problem Instanz  $\mathcal{T}$  mit 50 Jobs. Dabei wird die Bearbeitungszeit aus einer Gleichverteilung mit  $p_j = \lfloor \mathcal{U}(1, 10) \rfloor$ ,  $\forall j = 1 \dots n$  erzeugt. Damit jedes Fälligkeitsdatum prinzipiell erreicht werden kann, werden diese Werte gleichverteilt, basierend auf  $p_j$ , erzeugt:  $d_j = p_j + \lfloor \mathcal{U}(1, 990) \rfloor$ ,  $\forall j = 1 \dots n$ . Die erzeugte Instanz ist in Tabelle A.12 im Anhang A.6 zu finden. Die Anwendung des optimalen Algorithmus liefert eine Paretofront bestehend aus 34 optimalen Lösungen. Diese sind im gleichen Anhang in Tabelle A.13 zu finden.

Für die Evaluation des Räuber-Beute-Modells verwenden wir eine feste Räuberkonfiguration, die nicht bezüglich der Parametereinstellungen optimiert wurde. Ebenso verzichten wir an dieser Stelle auf eine statistisch vollständige Auswertung. Vielmehr konzentrieren wir uns auf Beobachtungen, die eine Beschreibung des Systemverhaltens erlauben. Grundsätzlich gilt, dass alle hier getätigten Aussagen auf vielen Wiederholungen des gleichen Experiments basieren.

TABELLE 10.1: *Parametrisierung der Räuber für die Evaluation des Einmaschinenproblems  $1||\sum C_j, L_{max}$  sowohl für die Einzelbetrachtung als auch die gemeinsame Betrachtung der Kriterien.*

Räuber	Kriterium	Variation	Parameter
R1	$L_{max}$	EDD	$\sigma = 4$
R2	$\sum C_j$	EDD	$\sigma = 4$
R3	$L_{max}$	SPT	$\sigma = 4$
R4	$\sum C_j$	SPT	$\sigma = 4$

Wir wählen vier Räuber mit unterschiedlicher Konfiguration, siehe Tabelle 10.1. Für jedes Kriterium existieren zwei Räuber, von denen jeweils einer mit einer SPT-Variation und der andere mit einer EDD-Variation ausgestattet wird. Damit sind wir in der Lage, sowohl den Effekt der Verstärkung wie auch die lexikographische Ordnung der Kriterieneinflüsse zu betrachten.

Unsere Evaluation beschränken wir auf insgesamt 6000 Funktionsauswertungen

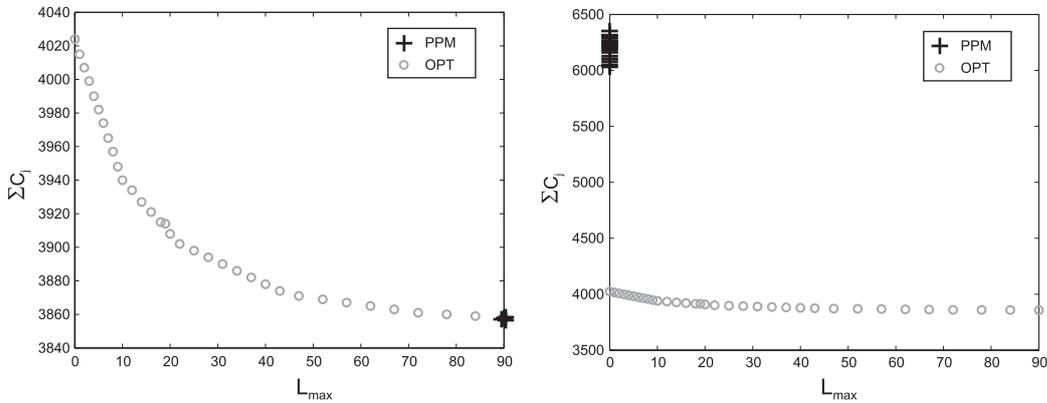


ABBILDUNG 10.3: Anwendung des Räuber-Beute-Ansatzes auf das Problem  $1||\sum C_j, L_{max}$  mit nur einem Räuber für  $\sum C_j$  (links) und  $L_{max}$  (rechts). Die Lösungen des Räuber-Beute-Modells sind mit + gekennzeichnet, während die optimale Lösung durch die o dargestellt werden. Die Räuber sind im Sinne der Verstärkung jeweils mit einem ihr Kriterium unterstützenden Variationsoperator verbunden (SPT bzw. EDD).

und betrachten zuerst die Einflüsse aller Räuberkonfigurationen einzeln. Abbildung 10.3 stellt die Ergebnisse für die ausschließliche Anwendung von R4 (links) und R1 (rechts) dar. Ganz offensichtlich konvergiert die Lösung durch den Einsatz der das Kriterium unterstützenden Variation in die Optimallösung des jeweiligen Teilproblems. Dieser Effekt der Verstärkung ist nicht überraschend, da sich zwangsläufig durch Anwendung der Sortierung (ähnlich wie bei Bubble-Sort (Knuth, 1997)) schließlich die optimale Sequenz einstellt.

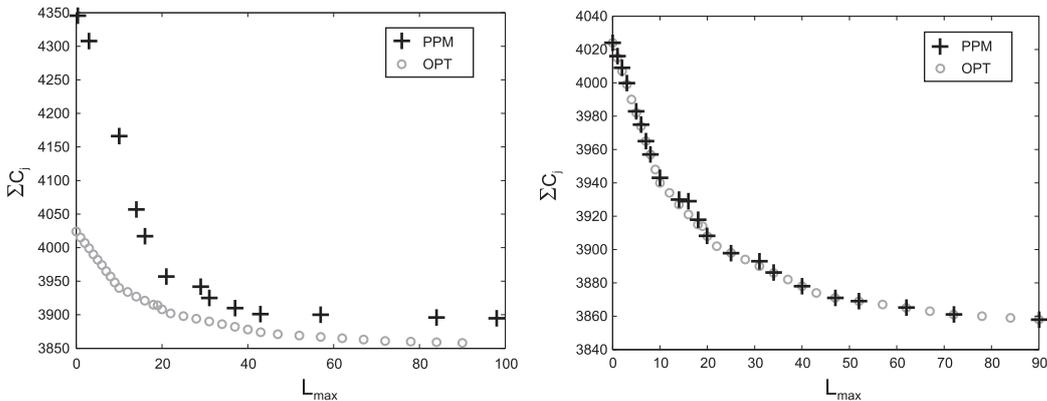


ABBILDUNG 10.4: Anwendung des Räuber-Beute-Ansatzes auf das Problem  $1||\sum C_j, L_{max}$  mit nur einem gemäß  $L_{max}$  selektierenden Räuber und einem Operator mit SPT-Tendenz (links). Die rechte Seite stellt die Anwendung aller Räuber gemeinsam dar. Die Lösungen des Räuber-Beute-Modells sind mit + gekennzeichnet, während die optimale Lösung durch die o dargestellt werden.

Abbildung 10.4 zeigt das Ergebnis der ausschließlichen Anwendung von R3 (links) und der gemeinsamen Anwendung aller Räuber (rechts). Bei Anwendung

der Selektion nach  $L_{max}$  und SPT-Variation entsteht ein zuerst überraschendes Ergebnis, das jedoch – im Sinne der lexikographischen Ordnung beider Kriterien – mit unserer Vermutung bezüglich des Zusammenwirkens von Räuber und Variation übereinstimmt. Der Räuber ist einerseits in der Lage, die optimale Lösung bezüglich seines Kriteriums aufzufinden. Gleichzeitig schafft es die Variation aber auch, ein Gegengewicht zu setzen und eine Kontraktion der Lösungsmenge – wie durch den Effekt der Verstärkung in Abbildung 10.3 hervorgerufen – zu verhindern. Es werden sogar Lösungen nahe an der optimalen Front erzeugt. Der Vollständigkeit halber sei erwähnt, dass ein ähnlicher aber wesentlich weniger ausgeprägter Effekt auch für Räuber R2 beobachtet werden kann. Hier findet allerdings kaum Annäherung an die Paretofront statt. Die Diversität wird jedoch gut erhalten. Diese Beobachtung bestätigt unsere Überlegungen zu Auswirkungen der verschiedenen Kombinationen von Räubern (also Selektion) und Variationsoperatoren.

Die Betrachtung des Gesamtergebnisses aus der gemeinsamen Anwendung aller Räuber bestätigt, dass das eingesetzte Modell durch die Kombination spezieller heuristischer Einflüsse in der Lage ist, die optimale Paretofront weitgehend zu erreichen. Es sei hier darauf hingewiesen, dass diese Ergebnisse nur mit der Anwendung von auf Heuristiken basierenden Operatoren und ohne Einsatz ungerichteter Mutation entstanden sind. Diese Ergebnisse werden wir als guten Ausgangspunkt, um das Potential des Räuber-Beute-Modells auf schwereren Schedulingproblemen genauer zu untersuchen.

### 10.3 Anwendung auf Mehrmaschinenprobleme

Nach der eben durchgeführten Vorstudie wollen wir nun Probleme mit mehreren Maschinen betrachten. Dabei untersuchen wir, ob trotz der Steigerung der Problemkomplexität eine Kombination einfacher Heuristiken im Räuber-Beute-Modell zum Erfolg führt. Wir beginnen mit zwei Problemen, innerhalb derer nur eines der beiden Teilprobleme NP-schwer ist. Danach erhöhen wir die Komplexität durch Hinzufügen von Einreichungszeiten. Abschließend untersuchen wir ebenfalls die Erhöhung der Anzahl verfolgter Ziele. Diese Betrachtungen erlauben uns eine umfassende Beurteilung der Fähigkeiten des Räuber-Beute-Modells im Kontext kombinatorischer Probleme und gestatten uns zudem eine Aussage über die Fähigkeit des Modells, verfügbares Problemwissen zu integrieren.

#### 10.3.1 Probleminstanzen, ein Vergleichsalgorithmus und Auswertungsvorgehen

Für die im Folgenden betrachteten Probleme mit parallelen Maschinen, legen wir bereits zu Beginn einige Randbedingungen fest. Um eine umfassende Untersuchung der Probleme durchzuführen, definieren wir Testinstanzen bestehend aus mehreren Maschinenkonfigurationen und 90 synthetischen Job-Mengen. Diese 90 Mengen können in drei Sätze unterteilt werden. Der erste und zweite Satz von Probleminstanzen enthält mit  $(\mathcal{J}_1^{50} \dots \mathcal{J}_{30}^{50})$  und  $(\mathcal{K}_1^{50} \dots \mathcal{K}_{30}^{50})$  dreißig Jobmengen bestehend

aus je 50 Jobs. Der dritte Satz ( $\mathcal{J}_1^{100} \dots \mathcal{J}_{30}^{100}$ ) besteht ebenfalls aus dreißig Jobmengen mit jeweils 100 Jobs. Wir erzeugen alle Mengen mit zufälliger Charakteristik gemäß  $p_j = \lfloor \mathcal{U}(1, 50) \rfloor$ ,  $\forall j = 1 \dots n$  und  $d_j = p_j + \lfloor \mathcal{U}(1, 100) \rfloor$ ,  $\forall j = 1 \dots n$ . Für die Probleminstanzen  $\mathcal{K}_i^{50}$ ,  $i = 1, \dots, 30$  erzeugen wir gemäß

$$r_j = \begin{cases} \mathcal{U}(0, d_j - p_j) & , \mathcal{U}(0, 1) \leq 0.9 \\ 0 & , \text{sonst} \end{cases}$$

Einreichungszeitpunkte, für die spätere Nutzung in Abschnitt 10.4. Weiterhin betrachten wir für den ersten und dritten Satz von Probleminstanzen jeweils eine Maschinenkonfiguration mit 8 und 12 Maschinen.

Wie bereits in Kapitel 9 beschrieben, liegen in der Literatur keine dedizierten heuristischen Ansätze vor, um die im Folgenden betrachteten Mehrmaschinenproblem (approximativ) zu lösen. Zwar können wir die optimale Lösung von  $P_k || \sum C_j$  mittels SPT exakt bestimmen und die Teilprobleme  $P_k || C_{max}$  mit LPT, sowie  $P_k || \sum U_j$  mit SBC3 gut approximieren. Für die Approximation der gesamten Paretofront sind wir jedoch auf generelle Heuristiken angewiesen. Um eine Referenzlösung zu erreichen, verwenden wir NSGA-2, da dieser Ansatz in der Literatur vielfach als Standardansatz zur mehrkriteriellen evolutionären Optimierung dargestellt wird (Coello et al., 2007). Außerdem existieren von diesem Algorithmus gute und leicht wiederverwendbare Standardimplementierungen, die hier auf die Schedulingproblematik angepasst werden können (Durillo, Nebro & Alba, 2010; Seshadri, 2009).

Für einen tatsächlich fairen Vergleich von Ergebnissen beider Algorithmen ist es notwendig, eine möglichst gute Parametrisierung von NSGA-2 vorzunehmen. Dies bedeutet insbesondere auch, zu versuchen, das Expertenwissen, das in das Räuber-Beute-Modell integriert wird, ebenso in NSGA-2 aufzunehmen. In verschiedenen Vorstudien wurden Einstellungen abweichend von den von Deb et al. vorgeschlagenen Konfigurationen (Deb et al., 2002) getestet. Außerdem wurde versucht, das Problemwissen über die betrachteten Teilprobleme auf analoge Weise wie im Räuber-Beute-Modell (also über die Variationsoperatoren), in NSGA-2 zu integrieren. Alle Versuche ergaben kein besseres Ergebnis als die vorgeschlagene Standardkonfiguration. Aus diesem Grunde wird im Folgenden die Standardkonfiguration von NSGA-2 zur Erzeugung von Referenzlösungen verwendet. Als Mutationsoperator wird im Kontext der Schedulingprobleme eine ungerichtete Vertauschungsmutation (Swap Mutation,  $\sigma = 8$  Vertauschungen) genutzt. Außerdem wird die Populationsgröße von NSGA-2 auf 100 Individuen begrenzt. Um eine Vergleichbarkeit anhand der algorithmischen Schritte zu erreichen, wurde ein Maximum von 6.000 Funktionsauswertungen festgelegt.

Für die Bewertung der Ergebnisse der folgenden Untersuchungen verwenden wir wie bei allen vorherigen Untersuchungen innerhalb dieser Arbeit das Hypervolumen von Zitzler & Thiele (1999). Entsprechende Referenzpunkte für die Berechnung werden in jedem Abschnitt angegeben. Da die Auswertungsmethode

unabhängig von der Kenntnis der optimalen Lösung ist, können wir so die eingeschlossenen Volumina von Lösungen aus der Anwendung des NSGA-2 und des Räuber-Beute-Modells (in der Auswertung mit PPM abgekürzt) vergleichen, um eine Qualitätsaussage über die algorithmische Leistungsfähigkeit bezüglich Approximationsqualität und Diversität der Lösungsmengen zu erhalten. Grundsätzlich überprüfen wir auch hier alle statistischen Aussagen mit dem Wilcoxon-Test ( $p < 0.05$ ) auf signifikante Unterschiede.

Eine andere Perspektive auf die Lösung und ein Maß zum Vergleich von Lösungsmengen liefert der additive  $\varepsilon$ -Indikator, siehe ebenfalls Zitzler & Thiele (1999). Der Indikator wird mit  $I_\varepsilon(A, B)$  bezeichnet und gibt an, ob eine Menge  $A$  eine andere Menge  $B$  vollständig dominiert. Genauer gesagt, gibt die additive Variante an, um wieviele Einheiten im Bildraum die Menge  $B$  insgesamt (in alle Koordinatenrichtungen) verschoben werden muss, um die Menge  $A$  zu dominieren. Für eine vollständige Aussage über die gegenseitige Dominanzbeziehung der Mengen  $A$  und  $B$  ist es daher notwendig, die Metrik in beide Richtungen anzuwenden. Nur wenn  $I_\varepsilon(A, B) > 0$  und  $I_\varepsilon(B, A) \leq 0$  gilt, dominiert die Menge  $A$  die Menge  $B$  vollständig. Andernfalls gibt es eine Überschneidung der beiden Mengen oder  $B$  dominiert sogar  $A$ . Für unsere Auswertung verwenden wir den Indikator binär. Wenn die obige Bedingung eintritt, vermerken wir dies als Dominanz der Menge  $A$  über  $B$ . Den umgekehrten Fall der Dominanz der Menge  $B$  über  $A$  vermerken wir ebenfalls. Daraus ergibt sich schließlich eine Anzahl an Dominanzbeziehungen, die uns angibt, wie oft ein Algorithmus bessere Lösungen als ein anderer Algorithmus erzeugt.

### 10.3.2 Das Problem $P_k || C_{max}, \sum C_j$

Für die Lösung des Problems  $P_k || C_{max}, \sum C_j$  nutzen wir zwei abgeleitete Varianten von dem in Abschnitt 10.1.2 definierten allgemeinen Konzept für Variationsoperatoren: Die sogenannte SPT-Mutation (SPTMut) ist abgeleitet von der SPT-Regel und ist vorteilhaft für das Problem  $P_k || \sum C_j$ . Die LPT-Mutation (LPTMut) ist eine gute Heuristik für das Teilproblem  $P_k || C_{max}$ . Wir konstruieren vier Räuber, von denen jeweils zwei das gleiche Kriterium verfolgen. Diese beiden Räuber werden jeweils mit den Operatoren SPTMut ( $\sigma = 5$ ) und LPTMut ( $\sigma = 10$ ) ausgestattet. Die Mutationsschrittweite  $\sigma$  wird jeweils auf Grundlage vorheriger manueller Parameterstudien gewählt.<sup>1</sup> Hier verzichten wir auf eine rein zufällige Mutation. Die Räuber bewegen sich auf einem  $10 \times 10$ -Torus, also auf einer Population mit 100 Individuen, maximal 6000 Funktionsauswertungen lang. Wir wiederholen jedes Experiment mindestens 30 mal, um eine statistisch tragfähige Aussage zu treffen.

In Abbildung 10.5 sind beispielhaft Ergebnisse der Auswertung für das Räuber-Beute-Modell und NSGA-2 dargestellt. Der linke Teil der Abbildung zeigt, dass

<sup>1</sup>In den durchgeführten Vorstudien wurden verschiedene Parameterkombinationen wiederholt evaluiert und „gute“ Ergebnisse für die angegebenen Werte erreicht. Eine systematische Evaluation des Parameterraumes wurde nicht vorgenommen, da hier die allgemeine Anwendbarkeit des Prinzips im Vordergrund der Betrachtung steht.

TABELLE 10.2: Referenzpunkte für die Hypervolumenberechnung.

$n$	$k$	Referenzpunkt
50	8/12	[1000, 5000]
100	8/12	[1000, 20000]

die vom Räuber-Beute-Modell ermittelte Lösungsmenge für eine Instanz des Problems  $P8||C_{max}, \sum C_j$  alle Lösungen des Vergleichsalgorithmus NSGA-2 dominiert. Ebenfalls ist in der Abbildung zu erkennen, dass im Falle des Einzelziels  $\sum C_j$  die optimale SPT-Lösung erreicht und im Falle von  $C_{max}$  die von LPT erzeugte Lösung übertroffen wird. Die Ergebnisse des Hypervolumens für alle Konfigurationen mit 50 bzw. 100 Jobs der Instanzen  $\mathcal{J}_i^{50}$  und  $\mathcal{J}_i^{100}$  sowie aller Maschinenkonfigurationen mit 8 und 12 Maschinen sind aus Platzgründen nicht dargestellt, bestätigen aber in der Gesamtbetrachtung die Feststellung, dass das Räuber-Beute-Modell über alle Experimente ein signifikant größeres Hypervolumen einschließt als NSGA-2. Für die Auswertung wurden die in Tabelle 10.2 angegebenen Referenzpunkte verwendet.

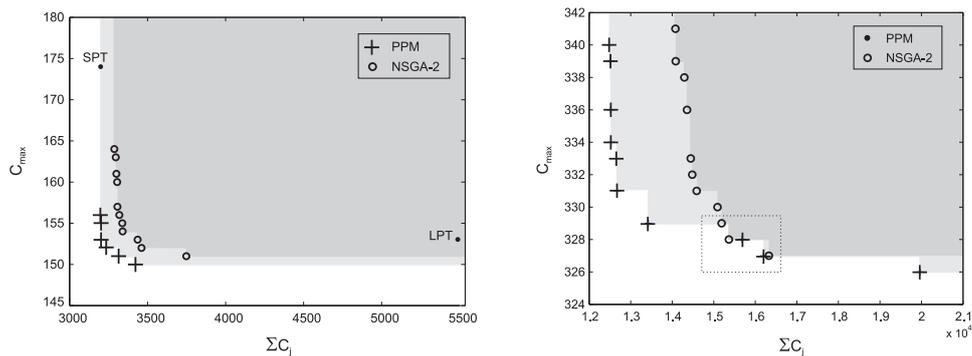


ABBILDUNG 10.5: Anwendung des Räuber-Beute-Ansatzes auf die Instanzen  $\mathcal{J}_1^{50}$  und  $\mathcal{J}_{15}^{50}$  des Problems  $P8||\sum C_j, L_{max}$ . Rechts ist eine Situation dargestellt, in der der Ansatz die NSGA-2 Lösung nicht vollständig dominiert. Die Lösungen des Räuber-Beute-Modells sind mit + gekennzeichnet, während die optimale Lösung durch die o dargestellt werden. Zusätzlich und zum Vergleich sind die Lösungen von SPT und LPT in der linken Abbildung eingezeichnet.

Zusätzlich fasst Tabelle 10.3 die Resultate bezüglich des  $\varepsilon$ -Indikators für alle Instanzen und Maschinenkonfigurationen zusammen. Die Betrachtung des  $\varepsilon$ -Indikators zeigt, dass das Räuber-Beute-Modell den NSGA-2 in den meisten Fällen übertrifft, sogar oftmals die gesamte Lösungsmenge von NSGA-2 dominiert. Mit zunehmender Jobanzahl in einer Instanz wird die Leistung des Räuber-Beute-Modells im Vergleich zu NSGA-2 besser. Gleichzeitig können wir feststellen, dass NSGA-2 in keinem Fall eine bessere Lösung als das Räuber-Beute-Modell generiert hat. Trotzdem müssen wir festhalten, dass in einigen Fällen das Räuber-Beute-Modell keine die NSGA-2-Lösung vollständig dominierende Lösungsmenge erzeugt. Bei genauerer Betrachtung, siehe auch Abbildung 10.5 rechts, dominie-

ren einzelne Lösungen in der Lösungsmenge von NSGA-2 einzelne Lösungen in der Lösungsmenge des Räuber-Beute-Modells. Dies führt zu einer Abwertung in unserer Metrik. Ein Grund für diesen Effekt können wir in der Komplexität des Zieles  $C_{max}$  vermuten. Während die SPT-Sortierung in SPTMut die Annäherung an die Paretofront direkt unterstützt, garantiert LPT als heuristischer Ansatz nicht das Erreichen der optimalen Lösung für  $C_{max}$ . Dies kann zu einem langsameren Fortschritt der Lösungsbildung in diesem Kriterium führen.

TABELLE 10.3: *Resultate für die Auswertung des  $\varepsilon$ -Indikators für die Probleme  $P8||C_{max}, \sum C_j$  und  $P12||C_{max}, \sum C_j$ . Der Indikator gibt Auskunft über die Dominanzbeziehung zweier Lösungsmengen. Der Operator  $A \triangleright_\varepsilon B$  bezeichnet die prozentuale Dominanz von A über B bezüglich der  $\varepsilon$ -Dominanz. Die Spalten  $PPM \triangleright_\varepsilon NSGA-2$  und  $NSGA-2 \triangleright_\varepsilon PPM$  zeigen (in %) die Resultate des Vergleichs von PPM und NSGA-2 bzw. andersherum für alle Jobinstanzen  $\mathcal{J}_1^{50} \dots \mathcal{J}_{30}^{50}$  und  $\mathcal{J}_1^{100} \dots \mathcal{J}_{30}^{100}$  sowie beide Maschinenkfigurationen.*

Jobinstanzen	$k$	PPM $\triangleright_\varepsilon$ NSGA-2		NSGA-2 $\triangleright_\varepsilon$ PPM
		% (Mittel)	std.	%
$\mathcal{J}_1^{50} \dots \mathcal{J}_{30}^{50}$	8	72.76	16.31	0
$\mathcal{J}_1^{50} \dots \mathcal{J}_{30}^{50}$	12	91.66	6.56	0
$\mathcal{J}_1^{100} \dots \mathcal{J}_{30}^{100}$	8	56.54	12.32	0
$\mathcal{J}_1^{100} \dots \mathcal{J}_{30}^{100}$	12	100	0.02	0

Grundsätzlich können wir aber sicher nach den ersten Versuchen die Erkenntnis festhalten, dass sich das Räuber-Beute-Modell inklusive der entwickelten Operatoren gut auf das Problem paralleler Maschinen mit verschiedenen Zielstellungen anwenden lässt. In ähnlicher Weise werden wir nun ein weiteres Problem untersuchen.

### 10.3.3 Das Problem $P_k||\sum C_j, \sum U_j$

Die Untersuchung des Problems  $P_k||\sum C_j, \sum U_j$  umfasst insbesondere die NP-schwere Problemstellung  $P_k||\sum U_j$ , während das Kriterium  $\sum C_j$  wiederum gut mit SPT zu lösen ist. Somit verwenden wir für  $\sum C_j$  erneut den SPTMut-Operator ( $\sigma = 2$ ). Für das Ziel  $\sum U_j$  wissen wir, dass die SBC3-Heuristik einen guten Lösungsansatz liefert. Eine Übertragung des Ansatzes in unseren sortierungs-basierten Variationsoperator ist jedoch nicht direkt möglich. Daher untersuchen wir zuerst, wie die Leistung von SBC3 als Kombination von einfachen Einflüssen angenähert werden kann.

Wir beginnen mit der Überlegung, dass SBC3 mit einer EDD entsprechenden Sequenzierung beginnt und sukzessive lange Jobs, die die Fälligkeitsdaten verletzen an das Ende der Sequenz sortiert. Da EDD eine auf den Fälligkeitsdaten basierende Sortierung ist, können wir dies leicht in einen entsprechenden Variationsoperator umwandeln. Ein zweiter Einfluss ergibt sich aus der Anwendung von SPTMut für das Kriterium  $\sum C_j$ . Durch diesen Operator wandern lange Jobs tendenziell an das

Ende der Sequenz. Dies entspricht grundsätzlich den beiden zuvor identifizierten Mechanismen, die in SBC3 wirken. Wir überprüfen nun zuerst die Approximation von SBC3-Ergebnissen durch eine Kombination von SPTMut, EDDMut und einem rein zufälligen Einfluss.

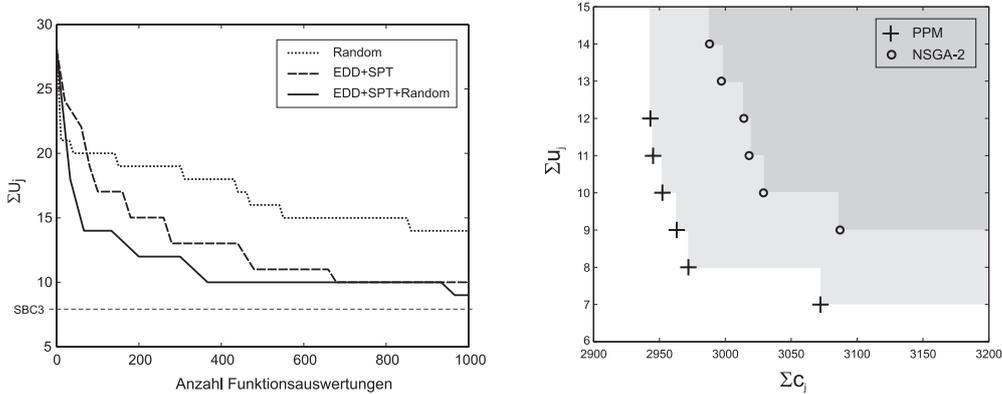


ABBILDUNG 10.6: Anwendung des Räuber-Beute-Ansatzes auf das Problem  $P8 \parallel \sum U_j$  zur Bewertung des Einflusses verschiedener Variationsoperatorkombinationen, links. Rechts ist beispielhaft ein Ergebnis der Anwendung des Ansatzes auf dem Problem  $P8 \parallel \sum C_j, \sum U_j$  dargestellt.

Das Experiment wird für das Problem  $P_k \parallel \sum U_j$  beispielhaft auf der Jobinstanz  $\mathcal{K}_1^{50}$  durchgeführt. Abbildung 10.6 zeigt links die Ergebnisse in der Zusammenfassung inklusive des durch SBC3 ermittelten Lösungswertes. Es wird ersichtlich, dass die Integration von SPT- und EDD-Sortierungen in den Variationsmechanismus durchaus vorteilhaft für das Konvergenzverhalten der einkriteriellen Suche ist. Dies gilt insbesondere im Vergleich zur ausschließlichen Anwendung von ungerichteter Mutation (Swap-Mutation, Random). Zudem erreichen wir bereits bei 1.000 Funktionsauswertungen annähernd die Lösungsqualität von SBC3. Wir schließen daraus, dass die Kombination von SPTMut, EDDMut und einem geringen ungerichteten Mutationseinfluss eine gute Möglichkeit für die Integration von Problemwissen aus dem SBC3-Verfahren ist, und verwenden dies nun im Folgenden für das eigentlich mehrkriterielle Problem.

Wir verwenden für das Problem  $P_k \parallel \sum C_j, \sum U_j$  die Jobinstanzen aus  $\mathcal{K}_i^{50}$  und konstruieren sechs Räuber, von denen jeweils drei ein gemeinsames Kriterium verfolgen. Jeweils zwei Räuber unterschiedlichen Kriteriums werden mit SPTMut, EDDMut ( $\sigma = 2$ ) und ungerichteter Mutation (Random,  $\sigma = 4$ ) ausgestattet. Die Wahl der Räuberkonfiguration ist in der Anzahl der verwendeten Variationsoperatoren begründet. Für jedes Kriterium wird ein Räuber erzeugt, der genau einen der drei Variationsoperatoren trägt. Für jedes Kriterium wählen wir gleich viele Räuber, um einen gleichmäßigen Einfluss auf die Population zu erreichen. Ansonsten verwenden wir, wie bereits im vorherigen Abschnitt 10.3.2, einen  $10 \times 10$ -Torus und 6.000 Funktionsauswertungen. Abweichend vom vorherigen Experiment bestimmen wir das Hypervolumen mit dem Referenzpunkt [5500,50].

Die Darstellung in Abbildung 10.6 rechts stellt exemplarisch ein Beispiel er-

TABELLE 10.4: *Resultate für die Auswertung des  $\varepsilon$ -Indikators bzgl. des Problems  $P8 || \sum C_j, \sum U_j$ . Die Spalten zeigen (in %) die Resultate des Vergleichs von PPM und NSGA-2 bzw. andersherum für die Jobinstanz  $\mathcal{K}_1^{50} \dots \mathcal{K}_{30}^{50}$ . Der Operator  $A \triangleright_\varepsilon B$  bezeichnet die prozentuale Dominanz von A über B bezüglich der  $\varepsilon$ -Dominanz.*

Jobinstanzen	$k$	PPM $\triangleright_\varepsilon$ NSGA-2		NSGA-2 $\triangleright_\varepsilon$ PPM
		% (Mittel)	std.	%
$\mathcal{K}_1^{50} \dots \mathcal{K}_{30}^{50}$	8	64.54	25.25	0.00

mittelter Lösungen für das Räuber-Beute-Modell und den Vergleichsalgorithmus NSGA-2 dar. Wir können auch hier feststellen, dass das Räuber-Beute-Modell NSGA-2 in seiner Lösungsqualität klar bessere Ergebnisse liefert. Dies wird auch durch eine vollständige Analyse der Hypervolumenergebnisse bestätigt. Es gibt eine signifikante Qualitätssteigerung durch den Einsatz des Räuber-Beute-Modells mit integriertem Problemwissen. Auch die Betrachtung der Ergebnisse unter dem  $\varepsilon$ -Indikator in Tabelle 10.4 zeigt, dass die überwiegende Anzahl der vom Räuber-Beute-Modell erzeugten Lösungen die Lösungen von NSGA-2 vollständig dominieren. Trotzdem findet sich auch hier eine Anzahl von Lösungsmengen für das Räuber-Beute-Modell, die zumindest teilweise Überschneidungen mit Lösungsmengen von NSGA-2 aufweisen. Es ist aber auch klar, dass NSGA-2 in keinem Fall die Lösungsqualität des Räuber-Beute-Modells übertrifft. Insgesamt ist das Räuber-Beute-Modell also erfolgreich bei der Approximation einer guten Lösungsmenge.

## 10.4 Modularität und Erweiterbarkeit

Die Beobachtungen zu den beiden zweikriteriellen Problemen in den Abschnitten 10.3.2 und 10.3.3 zeigen auf, dass das Räuber-Beute-Modell grundsätzlich gute Lösungen für Schedulingprobleme finden kann. In den meisten Fällen gelingt dies besser als mit NSGA-2. Diese Überlegenheit resultiert natürlich aus der Integration von Expertenwissen über die Teilprobleme. Zugleich – und als Motivation für die Verwendung des Räuber-Beute-Modells von zentraler Bedeutung – hat die Untersuchung auch die Einfachheit der Integration von Expertenwissen belegt. Im Folgenden sollen beide Aspekte – die Überlegenheit gegenüber NSGA-2 als etabliertem Verfahren und die Integration von Problemwissen – noch etwas genauer betrachtet werden. Dazu erhöhen wir einerseits die Problemkomplexität, um detaillierter zu untersuchen, wie das Modell schrittweise um weiteres Problemwissen erweitert werden kann. Andererseits erhöhen wir die Anzahl der betrachteten Kriterien, um die Skalierbarkeit des Modells zu beurteilen.

### 10.4.1 Erhöhung der Problemkomplexität

Für die Erhöhung der Problemkomplexität betrachten wir das Problem aus Abschnitt 10.3.3 erneut. Wir verwenden die gleiche Eingabemenge, bestehend aus

30 Jobinstanzen, und weitgehend dieselben Parametereinstellungen. Das Problem gestalten wir schwerer, indem wir Einreichungszeitpunkte als Randbedingung hinzufügen. So entsteht das Problem  $P_k|r_j|\sum C_j, \sum U_j$ . Für beide Teilprobleme gilt nun, dass sie NP-schwer sind. Weder SPT noch SBC3 eignen sich damit noch als direkte Heuristiken zur Lösung der Teilprobleme. Zwar stellen sie noch brauchbare Ansätze dar, um sich dem Problem grundsätzlich zu nähern, die Einreichungszeitpunkte können jedoch die Lösungsqualität stark verzerren. Um diesem Effekt entgegen wirken zu können, entwickeln wir zusätzlich zu den bereits verfügbaren Variationsoperatoren SPTMut, EDDMut und Random einen weiteren Operator, der eine Sortierung bezüglich der Einreichungsdaten unterstützt (RDMut,  $\sigma = 3$ ). Zusätzlich zu der Anwendung aller Operatoren, gekoppelt an insgesamt 8

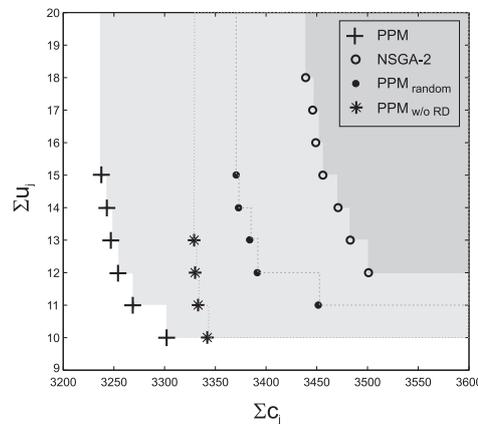


ABBILDUNG 10.7: Beispielhafte Darstellung der Ergebnisse für das Problem  $P8|r_j|\sum C_j, \sum U_j$  in verschiedenen Konfigurationen des PPM (● für reine Zufallsmutation, \* ohne Berücksichtigung eines Operators für Einreichungszeiten und + für die vollständige Konfiguration) sowie für NSGA-2 (Kreise).

Räuber (vier pro Kriterium), ermitteln wir Ergebnisse mit der Konfiguration aus Abschnitt 10.3.3, also ohne RDMut. Weiterhin evaluieren wir, wie sich das System ohne jegliches Problemwissen verhält. Als Vergleichsalgorithmus wird erneut NSGA-2 genutzt.

Beispielhaft sind Ergebnisse dieser Betrachtungen gemeinsam in Abbildung 10.7 dargestellt. Die statistisch detaillierte Auswertung des Hypervolumens ergibt eine signifikante Überlegenheit des Räuber-Beute-Modells gegenüber NSGA-2. Insbesondere unter dem Einsatz von Expertenwissen ist Räuber-Beute-Modell besser, als unter dem Einsatz ausschließlich ungerichteter Mutation. Zudem ist der positive Beitrag des neuen Operators RDMut deutlich nachzuweisen. Wir können also klar den Einfluss der Hinzunahme des zusätzlichen Problemwissens nachweisen. Gleichzeitig liefert uns diese Untersuchung ein weiteres wichtiges Ergebnis. Wir können zeigen, dass das Räuber-Beute-Modell auch ohne Problemwissen ein signifikant größeres Hypervolumen ermittelt als NSGA-2 dies tut.

Schließlich bestätigt auch die Betrachtung des  $\varepsilon$ -Indikators über alle Experi-

TABELLE 10.5: *Resultate für die Auswertung des  $\varepsilon$ -Indikators für das Problem  $P8 || \sum C_j, \sum U_j$ . Die Spalten zeigen (in %) die Ergebnisse des Vergleichs von PPM und NSGA-2 bzw. umgekehrt für die Jobinstanz  $\mathcal{K}_1^{50} \dots \mathcal{K}_{30}^{50}$ . Die Ergebnisse sind entsprechend der Notation in Abbildung 10.7 mit den verschiedenen Symbolen für die untersuchten Operatorkonfigurationen annotiert (• für reine Zufallsmutation, \* ohne Berücksichtigung eines Operators für Einreichungszeiten und + für die vollständige Konfiguration).*

Jobinstanzen	$k$	PPM $\triangleright_\varepsilon$ NSGA-2		NSGA-2 $\triangleright_\varepsilon$ PPM
		% (Mittel)	std.	%
$\mathcal{K}_1^{50} \dots \mathcal{K}_{30}^{50}$	8	96.42•	3,70•	0.01•
		92.55*	9.47*	0.03*
		98.94+	2.16+	0.00+

mente in Tabelle 10.5 diese Ergebnisse. Fast alle Lösungen – egal ob durch die ausschließliche Anwendung von ungerichteter Mutation oder durch den Einsatz von Problemwissen entstanden – dominieren die Ergebnisse von NSGA-2.

#### 10.4.2 Erhöhung der Kriterienanzahl

Als Abschluss der Untersuchungen hinsichtlich der Anwendung des Räuber-Beute-Modells, soll nun noch die Skalierbarkeit bezüglich der Kriterienanzahl untersucht werden. Dafür wird auf eine Kombination der vorherigen Probleme aus den Abschnitten 10.3.2 und 10.3.3 Bezug genommen. Wir betrachten also das Problem  $P_k || C_{max}, \sum C_j, \sum U_j$ . Dabei verwenden wir weitgehend die Einstellungen der vorherigen Experimente auf den Jobinstanzen aus  $\mathcal{J}_i^{50}$  und  $\mathcal{J}_i^{100}$  in einem 8- und 12-Maschinen-Szenario. Wir wählen SPTMut ( $\sigma = 5$ ), EDDMut ( $\sigma = 5$ ), LPT ( $\sigma = 10$ ) und Random ( $\sigma = 4$ ) als Operatoren und stattdessen jeweils zwei Räuber unterschiedlicher Kriterien mit jedem Operator aus. Dies ergibt eine Gesamtanzahl von 8 Räubern.

TABELLE 10.6: *Referenzpunkte für Hypervolumenberechnung.*

$n$	$k$	Referenzpunkt
50	8/12	[1000, 7000, 50]
100	8/12	[1000, 20000, 100]

Für die Evaluation des Hypervolumens geben wir die Referenzpunkte abhängig von der genutzten Jobinstanz in Tabelle 10.6 an. Da wir uns nun im dreidimensionalen Lösungsraum befinden, müssen diese angepasst werden. Die Auswertung des Hypervolumens zeigt, dass das Räuber-Beute-Modell in allen Instanzen dem NSGA-2 signifikant überlegen ist. In Abbildung 10.8 ist exemplarisch die Lösungsmenge einer Job-Instanz (mit 8 Maschinen) für das Räuber-Beute-Modell und NSGA-2 dargestellt. Der Übersichtlichkeit halber ist die Darstellung in drei Projektionen bezüglich jeder Zwei-Kriterien-Kombination der Ziele aufgeteilt worden.

Für jede Projektion sind die den betrachteten Kriterien entsprechenden nicht-dominierten Lösungen hervorgehoben. Die festgestellte Überlegenheit des Räuber-Beute-Modells gilt ebenfalls für die Betrachtung der  $\varepsilon$ -Dominanz in Tabelle 10.7. In fast allen Konfigurationen werden die NSGA-2-Lösungen von den Lösungen des Räuber-Beute-Modells dominiert. Nur in wenigen Fällen ist keine direkte Dominanzaussage zu treffen. Hingegen können wir feststellen, dass NSGA-2 in keinem Fall eine Lösung des Räuber-Beute-Modells dominiert.

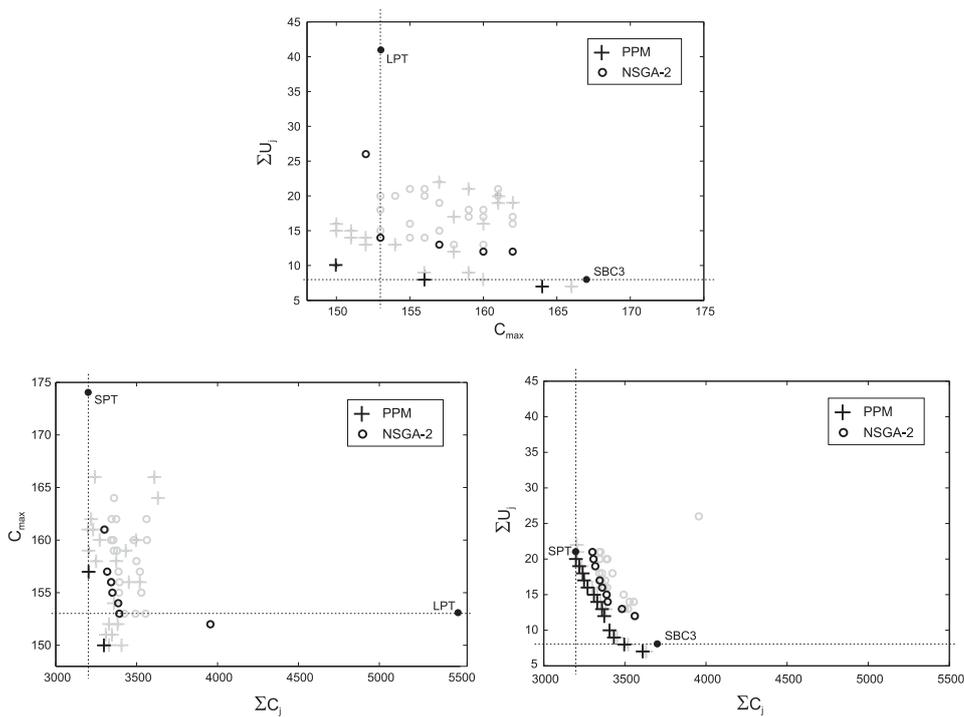


ABBILDUNG 10.8: Ergebnisse des Räuber-Beute-Modells auf dem Problem  $P8||C_{max}, \sum C_j, \sum U_j$ . Dargestellt sind die Projektionen bezüglich jeweils zweier Kriterien. In Grau sind jene Lösungen dargestellt, die zwar ebenfalls *Parto-optimal* bzgl. aller Kriterien sind, jedoch nicht zur *Paretofront* der projizierten Lösung gehören.

Es ist bemerkenswert, dass NSGA-2 gerade bei der Betrachtung von Instanzen mit 100 Jobs und 12 Maschinen in jedem Fall vom Räuber-Beute-Modell dominiert wird. Eine etwas genauere Betrachtung des Hypervolumens ergibt, dass die Differenz zwischen Räuber-Beute-Modell und NSGA-2 im normalisierten Hypervolumen oftmals annähernd 0.5, also fast die Hälfte des maximal erreichbaren Hypervolumens beträgt. Dies deutet darauf hin, dass NSGA-2 erhebliche Probleme bei der Konvergenz aufweist. Eine noch genauere Betrachtung offenbart, dass NSGA-2 insbesondere starke Probleme bei der Approximation der Kriterien  $\sum C_j$  und  $\sum U_j$  hat. In vielen Fällen ist das Räuber-Beute-Modell in der Lage, Lösungen ohne verspätete Jobs zu erzeugen (für 12 Maschinen), während NSGA-2 oftmals bei einer Größenordnung von 50 verspäteten Jobs stagniert. Ebenso kann NSGA-2

TABELLE 10.7: Resultate für die Auswertung des  $\varepsilon$ -Indikators und das dreikriterielle Problem  $P_k || C_{max}, \sum C_j, \sum U_j$ . Der Indikator gibt Auskunft über die Dominanzbeziehung zweier Lösungsmengen. Die Spalten zeigen (in %) die Resultate des Vergleichs von PPM und NSGA-2 bzw. umgekehrt für alle Jobinstanzen  $\mathcal{J}_1^{50} \dots \mathcal{J}_{30}^{50}$  und  $\mathcal{J}_1^{100} \dots \mathcal{J}_{30}^{100}$  sowie beide Maschinenkonfigurationen.

Jobinstanzen	$m$	PPM $\triangleright_\varepsilon$ NSGA-2		NSGA-2 $\triangleright_\varepsilon$ PPM
		% (Mittel)	std.	%
$\mathcal{J}_1^{50} \dots \mathcal{J}_{30}^{50}$	8	99.36	1.79	0
$\mathcal{J}_1^{50} \dots \mathcal{J}_{30}^{50}$	12	97.51	4.03	0
$\mathcal{J}_1^{100} \dots \mathcal{J}_{30}^{100}$	8	99.89	0.38	0
$\mathcal{J}_1^{100} \dots \mathcal{J}_{30}^{100}$	12	100	0.00	0

die Lösungen für  $\sum C_j$  nicht annähernd so gut approximieren wie das Räuber-Beute-Modell.

Wir können dieses Verhalten auf zwei Arten erklären: Erstens ist nach Knowles & Corne (2007) bekannt, dass NSGA-2 mit zunehmender Kriterienzahl ein schwächeres Konvergenzverhalten zeigt. Dies ist in der Selektion auf Grundlage der Nichtdominanz begründet. Sie konserviert Lösungen und versucht eine gute Diversität zu erzeugen. Hingegen wissen wir, dass das Räuber-Beute-Modell durch seine einkriterielle Selektion starken Druck bezüglich jedes Kriteriums ausübt. Dies kann zu einem guten Konvergenzverhalten beitragen. Zweitens steuert die Integration von Problemwissen einen wichtigen Teil zur Konvergenz bei.

## 10.5 Zusammenfassung und Diskussion

Nach der Betrachtung des Räuber-Beute-Modells im Kontext der mehrkriteriellen Schedulingprobleme, können wir verschiedene Erkenntnisse festhalten. Zuerst haben wir gezeigt, dass eine Übertragung des Modells in den Bereich der ganzzahligen Optimierung leicht möglich ist. Im Falle der Schedulingproblematik für Ein- und Mehrmaschinenprobleme haben wir eine einfache permutationsbasierte Repräsentation der Individuen gewählt. Gleichzeitig haben wir ein effektives Konzept zur Integration von Expertenwissen über die Variationsoperatoren entwickelt.

In diesem Kontext konnten wir die Leistungsfähigkeit des Räuber-Beute-Ansatzes genauer untersuchen. In einer ersten Studie auf dem polynomiell lösbaren Problem  $1 || \sum C_j, L_{max}$  ist es gelungen, die grundlegenden Einflüsse der verschiedenen Kombinationen von Räubern und problemspezifischen Variationsoperatoren nachzuweisen und schließlich die Nutzbarkeit des Ansatzes zu belegen. In einem zweiten Schritt der Anwendung auf schwerere Probleme im Themenbereich paralleler Maschinenumgebungen konnten wir ebenfalls zeigen, dass das Räuber-Beute-Modell gute Lösungen erstellt und – auch durch die Integration von Wissen über einkriterielle Resultate – NSGA-2 als Vergleichsansatz übertrifft. Die weiterführenden

de Untersuchung schwererer Probleme ergibt auch, dass das Räuber-Beute-Modell grundsätzlich leicht erweiterbar ist. So können über weitere Räuber und Operatoren zusätzliche Kriterien, aber auch weitere Randbedingungen in die Betrachtung aufgenommen werden.

Dieser letzte Punkt verdient für zukünftige Untersuchungen und die Nutzung des Ansatzes besondere Erwähnung. Im Gegensatz zu den meist monolithischen Ansätzen der mehrkriteriellen Optimierung erlaubt das Räuber-Beute-Modell eine einfache Skalierung, und zwar hinsichtlich der Kriterienanzahl und – noch wichtiger – der Problemkomplexität. Einkriterielles Wissen kann sehr einfach über die Kopplung von Räubern und Variationsoperatoren in den Algorithmus aufgenommen werden. Dabei muss sich ein Nutzer des Ansatzes nicht explizit um die Verschmelzung der einzelnen Einflüsse kümmern. Dies übernimmt das Räuber-Beute-Modell, indem es jeden Operator lokal wirken und zur Verbesserung der Gesamtlösung beitragen lässt. Aufgrund unserer bisherigen Erfahrung und des Verständnisses der Dynamik des Ansatzes können wir sogar spekulieren, dass auch die Hinzunahme falscher Einflüsse durch das Modell automatisch begrenzt wird. Es ist zu erwarten, dass ein Räuber, der mit einem ungeeigneten Einfluss auf die Population einwirkt, keine erfolgreiche Ersetzung von Beuteindividuen durchführen kann und somit auf lange Sicht auch keine Auswirkung auf die Population haben wird.

Damit erhalten wir quasi einen Bausteinansatz, der es erlaubt, geeignet erscheinendes Expertenwissen in das Lösungsverfahren aufzunehmen, ohne den algorithmischen Ansatz selbst verändern zu müssen.



# Zusammenfassung der Arbeit und Ausblick

Die nun abgeschlossene Arbeit hat im Kontext der mehrkriteriellen evolutionären Optimierung das von Laumanns, Rudolph und Schwefel bereits 1998 vorgeschlagene Räuber-Beute-Modell aufgegriffen. Dabei beruhte die Auswahl dieses in der Literatur nicht stark beachteten Ansatzes zuvorderst auf seiner wichtigsten Eigenschaft: seine Einfachheit in der Anwendung.

Das Modell repräsentiert jedes Kriterium eines mehrkriteriellen Problems als eigene Räuberspezies. Die Beuteindividuen hingegen repräsentieren Lösungskandidaten für das betrachtete Gesamtproblem und bilden eine auf einem toroidalen Graphen angeordnete Population, die von allen Räuberindividuen zugleich bedroht wird. Um diese Bedrohung zu erzeugen, bewegen sich die Räuber zufällig über die zu Grunde liegende Graphenstruktur und bringen einen einkriteriellen Evolutionsprozess in eine lokale Nachbarschaft ein. Allein die gleichzeitige Bedrohung – so die Erwartung – sollte zu einem Ausbilden von Kompromisslösungen für das mehrkriterielle Problem führen.

Der Wert dieses Modells für die Anwendung ergibt sich durch den sehr modularen Aufbau aus unabhängig agierenden Räubern und den eingesetzten, einfachen einkriteriellen evolutionären Verfahren. Zu Beginn der Arbeit haben wir dargestellt, dass dieser Ansatz ohne komplexe dominanzbasierte oder indikatorgetriebene Selektionsmechanismen auskommt, deren Einsatz teilweise nicht nur laufzeitintensiv ist, sondern oftmals auch eine erhebliche Komplexität in den Algorithmus einbringt. Zugleich, und auch diese Eigenschaft wurde in der Arbeit diskutiert, erlaubt die Modularität und das weitgehend entkoppelte Agieren der Räuber eine Parallelisierung der Ausführung des Algorithmus. Anders als in Verfahren mit panmiktischer Population und Generationenschranken bei der Auswertung von Individuengenerationen, erlaubt die lokale Selektion der Räuber auf kleinen Nachbarschaften eine fast vollständige Aufhebung dieser Schranken. Dies macht das Verfahren im Kontext realer Optimierungsaufgaben attraktiv, wenn sehr unterschiedliche Laufzeiten für Einzelauswertungen vorliegen.

Trotz dieser grundsätzlichen Vorteile des Modells, wurde bereits in dieser Arbeit vorausgehenden Studien erkannt, dass die vom Algorithmus gelieferte Approximationsqualität der Paretofront nicht kompetitiv mit etablierten mehrkriteriellen

evolutionären Verfahren ist. Wir konnten hier zudem feststellen, dass die vom Algorithmus gebildeten Lösungen oftmals ausschließlich Extremalösungen für einzelne Kriterien waren, sich Kompromisse aber nicht oder nur unzureichend ausbildeten. Weiterhin konnten wir durch die Einfärbung der Population (entsprechend der Qualität der Individuen) eine interessante Beobachtung machen: Die extrem ausgeprägten Beuteindividuen bilden in der Graphenstruktur feste Gebiete gleicher Qualität, die an den Grenzen scharf sind und keine Kompromissbildung aufweisen.

Ein erstes Ziel der Arbeit war es deshalb, eine detaillierte Analyse des Räuber-Beute-Modells durchzuführen, um einen Einblick in die Dynamik der Akteure und in den Prozess der Lösungsbildung zu erlangen. Dazu haben wir eine spezielle und leicht vereinfachte Variante des Räuber-Beute-Modells entwickelt, die – abweichend vom Original – die Gleichheit von Selektions- und Reproduktionsnachbarschaft für einen Räuber annimmt. Zudem haben wir den Auslöser für die Reproduktion und den genutzten Variationsoperator direkt an den Räuber gekoppelt. Dies erlaubte uns, zusammen mit der einkriteriellen Selektion, das gezielte lokale Einbringen und Untersuchen eines jeden Variationseinflusses. Die sich anschließende Analyse konzentrierte sich insbesondere auf das Räuberverhalten und die Entwicklung der Population unter verschiedenen Variationsoperatoren in lokalen Nachbarschaften.

Die Betrachtung des Mutationseinflusses konnte die beobachtete Tendenz des Verfahrens, stabile Gebiete bezüglich der von den Räubern verfolgten Kriterien zu schaffen, mit Hilfe eines Markow-Prozesses erklären und so eine theoretische Grundlage für weitere Folgerungen legen. So sind sicher die wichtigsten Ergebnisse dieser Untersuchung, dass die Gebiete gleicher Qualität unabhängig von der Mutationsschrittweite entstehen und zusätzlich gegen das kurzzeitige Eindringen von Räubern anderer Kriterien unempfindlich sind. Dies lässt den Schluss zu, dass eine Anpassung der Mutationsschrittweite für eine Veränderung des Modellverhaltens ungeeignet ist. Mit einer Untersuchung der Rekombination – insbesondere haben wir hier die Simplex-Rekombination als Ausprägung einer rotationsinvarianten intermediären Rekombination genutzt – konnten wir zeigen, dass eine konvexe Operation in der frühen Phase der Räuber-Beute-Interaktion zur Beschleunigung der Konvergenz beiträgt, schließlich aber zum Zusammenfallen der Lösungsmenge und so zum Diversitätsverlust führt. Erst die gemeinsame Anwendung von Mutation und Rekombination – jeweils von einzelnen Räubern unabhängig induziert – kann für Probleme mit konvexen Paretomengen zu einer guten Lösungsapproximation führen. In diesem Falle wirken die Extremaltendenz der Mutation und der zusammenziehende Einfluss der Rekombination gegen einander, so dass sich auf den Grenzen zwischen stabilen Extremalgebieten Zwischenlösungen ausbilden.

Insgesamt stellten diese Ergebnisse einen wichtigen Ausgangspunkt für zwei weitere Entwicklungsstränge im Rahmen dieser Arbeit dar: Einerseits war die Identifikation grundlegender Mechanismen Voraussetzung für exemplarische Erweiterungen des Räuber-Beute-Modells. Andererseits zeigte die indirekte Kombination verschiedener Operatoren über die Kopplung an unabhängig agierende Räuber einen Weg zur Ausnutzung der Modularität in späteren Anwendungen des An-

satzes auf. Auf beide Aspekte sind wir in den letzten beiden Teilen der Arbeit eingegangen.

Für die Erweiterung des Räuber-Beute-Modells haben wir zwei Ansätze evaluiert, die an ganz unterschiedlichen Elementen des Modells ansetzen:

1. Durch die Integration lokaler gradientenbasierter Suche in die Beuteindividuen, erreichten wir auch für Problemstellungen mit nicht-konvexer Paretomenge eine gute Approximation und zugleich eine Stabilisierung der erlangten Kompromisslösungen gegen den starken Räubereinfluss. Trotzdem konnten wir den globalen Räubereinfluss unverändert nutzen, um das Zielgebiet der global optimalen Lösungen im Suchraum zu ermitteln. Die lokale gradientenbasierte Suchstrategie trug in diesem Gebiet schließlich maßgeblich zur guten Approximation der gesamten Lösungsmenge bei. Dabei waren wir insbesondere in der Lage die Modularität und Parallelisierbarkeit des Räuber-Beute-Modells auszunutzen und die Einfachheit des Verfahrens selbst zu erhalten. Allerdings konnten wir auch feststellen, dass die Erweiterung die Leistungsfähigkeit des Räuber-Beute-Modells bezüglich der Approximationsqualität zwar steigerte, der Aufwand der Auswertung und die endgültige Ergebnisqualität jedoch weiterhin nicht kompetitiv zu etablierten Lösungen wie NSGA-2 und SMS-EMOA ist.
2. Der zweite Ansatz integrierte Elemente der  $\epsilon$ -Constraint Methode in die Räuberindividuen. Dabei schränkten wir den Zielfunktionenraum für alle von einem Räuber nicht verfolgten Ziele individuell ein. Die Überschreitung einer oder mehrerer oberer Grenzen für die  $m - 1$  beschränkten Kriterien durch eine Lösung führte zu einer Bestrafung und damit zur Abwertung der Lösung. Damit war es möglich, durch einen Räuber eine spezielle Region der Paretofront favorisieren zu lassen. Die Anwendung verschiedener Räuber mit individuellen Grenzen führte schließlich zu einer guten Abdeckung der Paretofront. Für einige Testprobleme konnten wir sogar zeigen, dass diese Erweiterung das Räuber-Beute-Modell kompetitiv zu etablierten Verfahren machte. Wichtig ist auch hier, dass die Modifikationen direkten Gebrauch von der Modell-eigenen Modularität machten. Insbesondere konnten wir die unabhängig agierenden Räuber durch lose Kooperation zur Bestimmung des Zielgebiets und zur automatischen Einstellung der Zielbeschränkungen für die integrierte  $\epsilon$ -Constraint Methode verwenden.

Beide Erweiterungen haben gezeigt, dass es mit Hilfe des erlangten Verständnisses der Dynamik des Räuber-Beute-Modells möglich ist, Erweiterungen zu entwickeln, die zuvor identifizierte Probleme beheben. Darüber hinaus haben wir mit beiden Ansätzen eindrucksvoll darstellen können, welches Potential in der Modularisierung des gesamten Modells liegt. Wir konnten durch die Manipulation zweier Module – nämlich von Räuberindividuen und Beuteindividuen – direkt eine Verbesserung des Verfahrens erreichen, ohne dabei die Komplexität des Modells

signifikant zu erhöhen. Die grundlegende Eigenschaft der Einfachheit konnte also bewahrt werden.

Das praktische Potential dieses Ansatzes konnten wir schließlich im letzten Teil der Arbeit mit der Übertragung des Räuber-Beute-Modells auf den Bereich der mehrkriteriellen Schedulingprobleme darstellen. In diesem Bereich ist zwar für viele Problemstellungen die Komplexität bekannt, gute Lösungsansätze oder heuristische Regeln gibt es aber fast ausschließlich für einkriterielle Probleme. In der Praxis ergibt sich oftmals das Problem, dass dieses einkriterielle Wissen nicht auf den mehrkriteriellen Fall übertragen werden kann. Mit Hilfe des Räuber-Beute-Modells konnten wir zeigen, dass eine solche Übertragung durch ein einfaches Baukasten-System zu erreichen ist. Indem wir die einkriteriellen Lösungsstrategien als Variationsoperatoren an die grundsätzlich einkriteriell arbeitenden Räuberindividuen binden, können wir diese sehr einfach als „Einfluss“ in die Entwicklung der Population einbringen. Wir haben dieses Prinzip in der Arbeit zuerst für das einfache und auch optimal lösbare Einmaschinenproblem  $1||\sum C_j, L_{max}$  gezeigt. Danach haben wir den Ansatz auf komplexere Probleme mit identischen parallelen Maschinen übertragen und dargestellt, dass dieses Verfahren bessere Ergebnisse als etablierte Ansätze wie NSGA-2 erzeugen kann. Bei seiner praktischen Anwendung spielt neben der Ergebnisqualität erneut die Einfachheit des Ansatzes bezüglich dreier Aspekte eine zentrale Rolle:

- *Skalierbarkeit bzgl. der Problemkomplexität:* Wir konnten durch die Steigerung von Problemkomplexität darstellen, dass der modulare Ansatz durch die einfache Hinzunahme von einkriteriellen, an neue Räuberspezies gebundene Heuristiken, auch die neue Problemstellung gut lösen kann.
- *Skalierbarkeit bzgl. der Kriterienanzahl:* Die Betrachtung weiterer Kriterien konnten wir durch die Konstruktion entsprechender zusätzlicher Räuberspezies abbilden.
- *Keine Notwendigkeit der Manipulation der Verfahrensstruktur:* Im Gegensatz zu vielen etablierten Verfahren ist es bei dem Einsatz des Räuber-Beute-Modells nicht nötig die algorithmische Struktur für die Integration von Expertenwissen (wie etwa die einfachen Scheduling-Heuristiken) zu verändern. Der modulare Ansatz von unabhängig agierenden Räubern und den Räubern zugeordneten Variationsoperatoren erlaubt das einfache Einbringen von Teilwissen für ein komplexes Problem.

Insbesondere der letzte Aspekt der vorausgehenden Aufzählung zeigt eine Perspektive für die Weiterentwicklung und Nutzung des Räuber-Beute-Modells auf. Durch die Möglichkeit der flexiblen Kombination heuristischen Wissens ergibt sich mittelfristig die Forderung, diesen Ansatz in Form eines algorithmischen Frameworks für die Lösung realer Problemstellungen einzusetzen. Dabei steht besonders der ingenieurtechnische Gedanke im Vordergrund, aus bekannten Teillösungen eine komplexe Lösungsstrategie zu entwickeln. Interessant wird der Ansatz zudem

durch die Tatsache, dass mit dem Räuber-Beute-Modell und über die Kopplung des Expertenwissens an einzelne Räuber, der Anwender – weitgehend unabhängig vom Verständnis der intern ablaufenden Prozesse des Modells – Bausteine seines Wissens zusammenfügen und quasi automatisch auf das tatsächliche Problem anwenden (lassen) kann. Für einen solchen Entwicklungsschritt ist die Übertragbarkeit des Modells auf weitere Problemklassen, für die ebenfalls einkriterielle Heuristiken vorliegen, zu prüfen. Zudem ergibt sich aus den in der Arbeit gelegten Grundlagen zur Dynamik des Modells die Annahme, dass die Kopplung einer für die Lösung unbrauchbaren Heuristik an einen Räuber kein Nachteil in der Entwicklung der Gesamtlösung ergibt. Wir vermuten vielmehr, dass eine kontraproduktive Variation zwar kurzzeitig lokale Veränderungen in die Population einbringen kann, diese aber direkt wieder vom Evolutionsprozess entfernt werden. So wird falsch eingebrachtes Wissen gleichsam ausgeschaltet. Eine genaue Untersuchung dieser Vermutung bleibt jedoch zukünftigen Arbeiten vorbehalten.

Insgesamt konnten wir mit der vorliegenden Arbeit sicherlich die Grundlagen für ein tieferes Verständnis der Prozesse innerhalb des Räuber-Beute-Modells legen und damit den Boden für zukünftige Weiterentwicklung bereiten. Exemplarisch haben wir zwei Ansätze gezeigt, die das Modell bereits verbessern und vor allem die Ausnutzung der Modularität des Verfahrens als viel versprechenden Ansatzpunkt identifizieren. Viel weitreichender und für die praktische Anwendung von hoher Relevanz ist schließlich die Untersuchung der Übertragung des Modells auf Schedulingprobleme. Hier ergibt sich eine wichtige Perspektive für den tatsächlichen ingenieurtechnischen Einsatz des Räuber-Beute-Modells.



# Anhang

## A.1 Zum Auffinden nicht-konvexer Lösungen durch die Gewichtungsmethode

Der folgende Beweis wurde von Das & Dennis (1997) vorgestellt und betrachtet die Problemformulierung aus Gleichung 3.1 für ein zweikriterielles MOP (siehe Gleichung A.1) als trigonometrische Formulierung:

$$\overline{MOP}_2^w = w_1 f_1(\vec{x}) + w_2 f_2(\vec{x}) \rightarrow \min \quad (\text{A.1})$$

$$\Leftrightarrow w_1 f_1(\vec{x}) + (1 - w_1) f_2(\vec{x}) \rightarrow \min, \text{ da } w_2 = 1 - w_1$$

$$\Leftrightarrow \frac{\cos \theta}{\cos \theta + \sin \theta} f_1(\vec{x}) + \frac{\sin \theta}{\cos \theta + \sin \theta} f_2(\vec{x}) \rightarrow \min \quad (\text{A.2})$$

mit  $\theta \in [0, \frac{\pi}{2}]$

Es gilt, dass wenn ein Pareto-optimaler Punkt keine Lösung für Gleichung A.2 ist  $\forall \theta \in [0, \frac{\pi}{2}]$ , dann kann dieser Punkt nicht durch eine Linearkombination der beiden Zielfunktionen erreicht werden. Weiterhin nehmen Das und Dennis an, dass die  $f_1$  und  $f_2$ -Achsen des Bildraumes gegen den Uhrzeigersinn im Winkel  $\theta$  gedreht werden. Es ergeben sich dann die neuen Achsen durch die in Gleichung A.3 durchgeführte Transformation.

$$\begin{pmatrix} \bar{f}_1 \\ \bar{f}_2 \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} \quad (\text{A.3})$$

Daraus resultiert, dass sich die Funktion  $\bar{f}_1(\vec{x}) = f_1(\vec{x}) \cos \theta + f_2(\vec{x}) \sin \theta$  ergibt und nach Einsetzen in A.2 der Ausdruck  $\frac{\bar{f}_1(\vec{x})}{\cos \theta + \sin \theta}$  minimiert werden muss. Es folgt direkt, dass nur  $\bar{f}_1(\vec{x})$  minimiert werden muss. Geometrisch liegt die Lösung also im globalen Optimum für  $\bar{f}_1(\vec{x})$ . Ein nicht konvexer Punkt stellt aber gerade kein globales Optimum dar, so dass dieser Punkt durch die Minimierung nicht gefunden wird.

□

## A.2 Die Definition der Probleme in der ZDT-Testsuite

TABELLE A.8: Formale Definition der sogenannten ZDT Testprobleme (Zitzler et al., 2000). Alle fünf Probleme sind reellwertige Probleme. ZDT5 ist hier ausgelassen worden, da es ein ganzzahliges Problem ist.

Name	Testproblem	Grenzen
ZDT1	$f_1(x) = x_1$ $f_2(x) = g(x) \cdot \left[ 1 - \sqrt{\frac{x_1}{g(x)}} \right]$ $g(x) = 1 + \frac{9}{n-1} \cdot \sum_{i=2}^n x_i^2$	$x_1 \in [0, 1]$ $x_i \in [-1, 1]$ $i = 2 \dots 30$
ZDT2	$f_1(x) = x_1$ $f_2(x) = g(x) \cdot \left[ 1 - \left( \frac{x_1}{g(x)} \right)^2 \right]$ $g(x) = 1 + \frac{9}{n-1} \cdot \sum_{i=2}^n x_i^2$	$x_1 \in [0, 1]$ $x_i \in [-1, 1]$ $i = 2 \dots 30$
ZDT3	$f_1(x) = x_1$ $f_2(x) = g(x) \cdot h(x)$ $g(x) = 1 + \frac{9}{n-1} \cdot \sum_{i=2}^n x_i^2$ $h(x) = \left[ 2 - \sqrt{\frac{x_1}{g(x)}} - \frac{x_1}{g(x)} \cdot \sin(10 \cdot \pi x_1) \right]$	$x_1 \in [0.1, 1]$ $x_i \in [-1, 1]$ $i = 2 \dots 30$
ZDT4	$f_1(x) = x_1$ $f_2(x) = g(x) \cdot \left[ 1 - \sqrt{\frac{x_1}{g(x)}} \right]$ $g(x) = 1 + 10 \cdot (n-1) + \sum_{i=2}^n (x_i^2 - 10 \cos(4\pi x_i))$	$x_1 \in [0, 1]$ $x_i \in [-5, 5]$ $i = 2 \dots 10$
ZDT6	$f_1(x) = 1 - \exp(-4x_1) \cdot \sin^6(4\pi x_i)$ $f_2(x) = g(x) \cdot \left[ 1 - \left( \frac{f_1(x)}{g(x)} \right)^2 \right]$ $g(x) = 1 + 9 \cdot \left[ \frac{\sum_{i=2}^n x_i^2}{n-1} \right]^{0.25}$	$x_1 \in [0, 1]$ $x_i \in [-1, 1]$ $i = 2 \dots 10$

### A.3 Evaluation der Gradientensuche in zehn Dimensionen

TABELLE A.9: Auswertungsergebnisse für die Einzelbetrachtung der gradientenbasierten Abstiegs-methode ohne Einfluss der Räuber. Die Ergebnisse beziehen sich auf 30 Wiederholungen eines Experimentes für das Problem  $\mathcal{F}_{\odot}^2$  im 10-dimensionalen Entscheidungsraum mit nur einem Individuum und kulminativer Betrachtung der Zwischenlösungen. Alle Werte sind Mittelwerte inkl. Standardabweichung.

Parameter	Initiale Phase
$r = 0.01,$ $t = 1$	$> 10000$
$r = 0.01,$ $t = 10$	$> 10000$
$r = 0.05,$ $t = 1$	$4437 \pm 223.2$
$r = 0.05,$ $t = 10$	$1219 \pm 49.4$
$r = 0.1,$ $t = 1$	$1910 \pm 132.8$
$r = 0.1,$ $t = 10$	$523 \pm 29.0$

### A.4 Referenzergebnisse etablierter Methoden für das Kursawe-Problem

TABELLE A.10: Evaluation mit Initialisierung in  $[-5, 5]^2$  und für die Anwendung von SMS-EMOA und NSGA-II bei 1000 Funktionsauswertungen.

Variante	HV	IGD
NSGA-II	$0.409 \pm 0.0028$	$2.27 \cdot 10^{-4} \pm 3.84 \cdot 10^{-5}$
SMS-EMOA	$0.410 \pm 0.0025$	$2.01 \cdot 10^{-4} \pm 3.07 \cdot 10^{-5}$

## A.5 Verwendete Konfigurationsparameter für die Evaluation des $\epsilon$ -Constraint-Ansatzes

TABELLE A.11: Konfiguration der jeweils 10 Räuber für die Auswertung des  $\epsilon$ -Constraint-Ansatzes integriert im Räuber-Beute-Modell.

Problem	Konfiguration	Problem	Konfiguration
Kursawe	2 R. $f_1$ , $\sigma = 0.2$	ZDT1	2 R. $f_1$ , $\sigma = 0.05$
	2 R. $f_2$ , $\sigma = 0.2$		2 R. $f_2$ , $\sigma = 0.05$
	2 R. $f_1$ , $\sigma = 0.01$		2 R. $f_1$ , $\sigma = 0.001$
	2 R. $f_2$ , $\sigma = 0.01$		2 R. $f_2$ , $\sigma = 0.001$
	1 R. $f_1$ , Simplex Rek., $r = 1$		1 R. $f_1$ , Simplex Rek., $r = 3$
	1 R. $f_2$ , Simplex Rek., $r = 1$		1 R. $f_2$ , Simplex Rek., $r = 3$
ZDT2	1 R. $f_1$ , $\sigma = 0.1$	ZDT3	1 R. $f_1$ , $\sigma = 0.1$
	3 R. $f_2$ , $\sigma = 0.1$		3 R. $f_2$ , $\sigma = 0.1$
	3 R. $f_1$ , $\sigma = 0.005$		2 R. $f_1$ , $\sigma = 0.005$
	1 R. $f_2$ , $\sigma = 0.005$		2 R. $f_2$ , $\sigma = 0.005$
	1 R. $f_1$ , Simplex Rek., $r = 2$		1 R. $f_1$ , Simplex Rek., $r = 2$
	1 R. $f_2$ , Simplex Rek., $r = 2$		1 R. $f_2$ , Simplex Rek., $r = 2$

## A.6 Instanz und Lösung für die Evaluation des Problems $1||\sum C_j, L_{max}$

TABELLE A.12: Eigenschaften der Problem Instanz  $\mathcal{T}$  für  $1||\sum C_j, L_{max}$ .

$j$	$p_j$	$d_j$									
1	10	793	14	5	170	27	3	526	40	7	100
2	1	827	15	1	158	28	7	859	41	10	645
3	1	50	16	4	522	29	2	571	42	10	776
4	1	484	17	9	533	30	8	393	43	5	190
5	3	513	18	3	829	31	2	610	44	7	10
6	6	69	19	1	167	32	2	156	45	9	530
7	3	824	20	3	928	33	3	632	46	10	362
8	1	951	21	5	292	34	1	357	47	7	487
9	7	537	22	8	391	35	2	849	48	1	800
10	1	90	23	2	151	36	3	785	49	8	466
11	1	968	24	9	302	37	6	192	50	5	749
12	6	702	25	5	279	38	2	678			
13	3	4	26	10	375	39	8	260			

TABELLE A.13: Größte Verspätung ( $L_{max}$ ) und Gesamtbearbeitungszeit ( $\sum C_j$ ) für alle 34 Pareto-optimalen Lösungen für die Problem Instanz  $\mathcal{T}$ .

$j$	$L_{max}$	$\sum C_j$									
1	0	4024	10	9	3948	19	25	3898	28	57	3867
2	1	4015	11	10	3940	20	28	3894	29	62	3865
3	2	4007	12	12	3934	21	31	3890	30	67	3863
4	3	3999	13	14	3927	22	34	3886	31	72	3861
5	4	3990	14	16	3921	23	37	3882	32	78	3860
6	5	3982	15	18	3915	24	40	3878	33	84	3859
7	6	3974	16	19	3914	25	43	3874	34	90	3858
8	7	3965	17	20	3908	26	47	3871			
9	8	3957	18	22	3902	27	52	3869			



# Literaturverzeichnis

- Ai, E. (2005). Ein Räuber-Beute-Ansatz für Evolutionsstrategien. Diplomarbeit Universität Dortmund, Fachbereich Informatik, Lehrstuhl für Systemanalyse.
- Alba, E. & Dorronsoro, B. (2008). *Cellular Genetic Algorithms*. Operation Research / Computer Science Interfaces. Springer.
- Alba, E. & Tomassini, M. (2002). Parallelism and Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 6(5), 443–462.
- Bäck, T. (1996). *Evolutionary Algorithms in Theory and Practice*. New York: Oxford University Press.
- Bäck, T., Fogel, D. B., & Michalewicz, Z. M. (1997). *Handbook of Evolutionary Computation*. New York: Oxford University Press.
- Bader, J. & Zitzler, E. (2009). A Hypervolume-Based Optimizer for High-Dimensional Objective Spaces. In *Conference on Multiple Objective and Goal Programming (MOPGP 2008)*, Lecture Notes in Economics and Mathematical Systems. Springer.
- Bader, J. & Zitzler, E. (2011). HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. *Evolutionary Computation*, 19(1), 45–76.
- Baker, J. E. (1985). Adaptive Selection Methods for Genetic Algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms*, (S. 101–111)., Hillsdale, NJ, USA. L. Erlbaum Associates Inc.
- Baluja, S. (1993). Structure and Performance of Fine-Grain Parallelism in Genetic Search. In *Proceedings of the 5th International Conference on Genetic Algorithms*, (S. 155–162)., San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Bartz-Beielstein, T., Lasarczyk, C., & Preuss, M. (2005). Sequential parameter optimization. In McKay, B. (Hrsg.), *Proceedings of the 2005 Congress on Evolutionary Computation*, Band 1, (S. 773–780)., Piscataway, New Jersey. IEEE Press.

- Beume, N. (2009). S-metric calculation by considering dominated hypervolume as klee's measure problem. *Evolutionary Computation*, 17(4), 477–492.
- Beyer, H.-G. (1995). Toward a theory of evolution strategies: On the benefits of sex—the  $(\mu/\mu, \lambda)$ -theory. *Evolutionary Computation*, 3, 81–111.
- Beyer, H.-G. (1997). An Alternative Explanation for the Manner in which Genetic Algorithms Operate. *BioSystems*, 41, 1–15.
- Beyer, H.-G. (2001). *The Theory of Evolution Strategies*. Berlin: Springer.
- Beyer, H.-G. & Schwefel, H.-P. (2002). Evolution strategies – A comprehensive introduction. *Natural Computing*, 1(1), 3–52.
- Binh, T. T. & Korn, U. (1996). An Evolution Strategy for the Multiobjective Optimization. In *Second International Conference on Genetic Algorithms*, (S. 23–26)., Brno, Czech Republic.
- Bosman, P. & Thierens, D. (2003). The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7(2), 174–188.
- Bosman, P. A. N. & de Jong, E. D. (2006). Combining Gradient Techniques for Numerical Multi-Objective Evolutionary Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, (S. 627–634)., New York, NY, USA. ACM.
- Bowman, V. J. J. (1976). On the Relationship of the Tchebycheff Norm and the Efficient Frontier of Multi-Criteria Objectives. In H. Thiriez & S. Zionts (Hrsg.), *Multiple Criteria Decision Making*, Lecture Notes in Economics and Mathematical Systems 130 (S. 76–85). Berlin: Springer.
- Brown, M. & Smith, R. E. (2005). Directed Multi-Objective Optimization. *International Journal of Computers, Systems and Signals*, 6, 3–17.
- Burke, E. K., Newall, J. P., & Weare, R. F. (1996). A Memetic Algorithm for University Exam Timetabling. In *Selected papers from the First International Conference on Practice and Theory of Automated Timetabling*, (S. 241–250)., London, UK. Springer-Verlag.
- Burke, E. K., Newall, J. P., & Weare, R. F. (1998). Initialization strategies and diversity in evolutionary timetabling. *Evolutionary Computation*, 6, 81–103.
- Campbell, N. A. & Reece, J. B. (2006). *Biologie*. Pearson.
- Cantú-Paz, E. (2005). Theory of Parallel Genetic Algorithms. In E. Alba (Hrsg.), *Parallel Metaheuristics: A New Class of Algorithms* (S. 425–445). John Wiley & Sons.

- Castro, L. R. d. & Timmis, J. (2002). *Artificial Immune Systems: A New Computational Intelligence Paradigm*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.
- Chen, C.-L. & Bulfin, R. L. (1993). Complexity of single machine multi-criteria scheduling problems. *European Journal of Operational Research*, 70, 115–125.
- Cheng, R. & Gen, M. (1996). Parallel machine scheduling problems using memetic algorithms. In *IEEE International Conference on Systems, Man, and Cybernetics, 1996.*, Band 4, (S. 2665–2670).
- Coello, C. A. C., van Veldhuizen, D. A., & Lamont, G. B. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems* (2. Aufl.). Genetic and Evolutionary Computation. NY: Springer.
- Coello Coello, C. A. & Lechuga, M. S. (2002). MOPSO: A proposal for multiple objective particle swarm optimization. In *Proceedings of the Conference on Evolutionary Computation, CEC '02*, (S. 1051–1056)., Washington, DC, USA. IEEE Computer Society.
- Dalgaard, P. (2002). *Introductory Statistics with R*. Statistics and Computing. New York: Springer.
- Darwin, C. (1859). *On the Origin of Species By Means of Natural Selection, or, The Preservation of Favoured Races in the Struggle for Life*. London: J. Murray, Nachdruck bei Dover (2006).
- Das, I. & Dennis, J. E. (1997). A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural and Multidisciplinary Optimization*, 14, 63–69.
- Dasgupta, D. & Attoh-Okine, N. (1997). Immunity-based systems: A survey. In *Proceeding of the IEEE International Conference on Systems, Man and Cybernetics*, (S. 369–374).
- Davis, L. (1985). Applying Adaptive Genetic Algorithms to Epistatic Domains. In *Proceedings of the International Joint Conference on Artificial Intelligence*, (S. 162–164).
- De Jong, K. A. (1975). *Analysis of the behavior of a class of genetic adaptive systems*. Dissertation University of Michigan at Ann Arbor, Michigan.
- Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley-Interscience Series in Systems and Optimization. Chichester: Wiley.
- Deb, K. & Goldberg, D. E. (1989). An Investigation of Niche and Species Formation in Genetic Function Optimization. In *Proceedings of the 3rd International Conference on Genetic Algorithms*, (S. 42–50)., San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197.
- Deb, K. & Udaya, B. R. N. (2005). Investigating Predator-Prey Algorithms for Multi-Objective Optimization. Technischer Bericht 2005010, Kanpur Genetic Algorithm Laboratory (KanGAL), Department of Mechanical Engineering, IIT, Kanpur, India.
- Dorigo, M. & Stützle, T. (2004). *Ant Colony Optimization*. Scituate, MA, USA: Bradford Company.
- Dreżewski, R. (2003). A model of co-evolution in multi-agent system. In *Proceedings of the 3rd Central and Eastern European conference on Multi-agent systems*, CEEMAS'03, (S. 314–323)., Berlin, Heidelberg. Springer-Verlag.
- Dreżewski, R. & Siwik, L. (2007). The Application of Agent-Based Co-Evolutionary System with Predator-Prey Interactions to Solving Multi-Objective Optimization Problems. In *IEEE Symposium on Computational Intelligence in Multicriteria Decision Making*, (S. 294 –301).
- Durillo, J., Nebro, A., & Alba, E. (2010). The jMetal Framework for Multi-Objective Optimization: Design and Architecture. In *IEEE Congress on Evolutionary Computation*, Band 5467, (S. 4138–4325)., Barcelona, Spain. Springer.
- Dutot, P.-F., Rzacca, K., Saule, E., & Trystram, D. (2010). Multi-objective scheduling. In Y. Robert & F. Vivien (Hrsg.), *Introduction to Scheduling* (1. Aufl.). (S. 219–251). CRC Press.
- Emmerich, M., Beume, N., & Naujoks, B. (2005). An EMO Algorithm Using the Hypervolume Measure as Selection Criterion. In C. Coello Coello, A. Hernandez Aguirre, & E. Zitzler (Hrsg.), *Evolutionary Multi-Criterion Optimization*, Band 3410 von *Lecture Notes in Computer Science* (S. 62–76). Springer.
- Ester, J. (1987). *Systemanalyse und mehrkriterielle Entscheidung*. Berlin: VEB Verlag Technik.
- Fogel, L. J., Owens, A. J., & Walsh, M. J. (1966). *Artificial Intelligence through Simulated Evolution*. New York, USA: John Wiley.
- Fonseca, C. & Fleming, P. (1995). Multiobjective genetic algorithms made easy: selection sharing and mating restriction. In *First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, 1995. GALEZIA.*, (S. 45–52).
- Fonseca, C. M. & Fleming, P. J. (1993). Genetic Algorithms for Multiobjective Optimization: Formulation Discussion and Generalization. In *Proceedings of the 5th International Conference on Genetic Algorithms*, (S. 416–423)., San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

- Franca, P. M., Mendes, A., & Moscato, P. (2001). A memetic algorithm for the total tardiness single machine scheduling problem. *European Journal of Operational Research*, 132(1), 224–242.
- Garey, M. R. & Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco: Freeman.
- Goldberg, D. E. (1983). *Computer-aided gas pipeline operation using genetic algorithms and rule learning*. Dissertation University of Michigan at Ann Arbor, Michigan.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning* (1st. Aufl.). Boston, MA, USA: Addison-Wesley Longman.
- Goldberg, D. E. & Deb, K. (1991). A comparative analysis of selection schemes in genetic algorithms. In G. J. E. Rawlins (Hrsg.), *Foundations of Genetic Algorithms 1* (S. 69–93). Morgan Kaufmann.
- Graham, R. L., Lawer, E. L., Lenstra, J. K., & Rinnooy Kan, A. H. G. (1979). Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, 5, 287–326.
- Grimme, C. (2005). Räuber-Beute-Systeme für die mehrkriterielle Optimierung. Interner Bericht der Forschungsgruppe Systemanalyse SYS-5/05, Technische Universität Dortmund.
- Grimme, C., Kemmerling, M., & Lepping, J. (2011). An expertise-guided multi-criteria approach to scheduling problems. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, (S. 47–48). ACM Press.
- Grimme, C. & Lepping, J. (2007). Designing Multi-Objective Variation Operators Using a Predator-Prey Approach. In *Proceedings of the Fourth International Conference on Evolutionary Multi-Criterion Optimization*, Band 4403 von *Lecture Notes in Computer Science*, (S. 21–35). Springer.
- Grimme, C. & Lepping, J. (2011a). Combining basic heuristics for solving multi-objective scheduling problems. In *IEEE Symposium on Computational Intelligence in Scheduling (SCIS)*, (S. 9–16).
- Grimme, C. & Lepping, J. (2011b). Integrating niching into the predator-prey model using epsilon-constraints. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, (S. 109–110). ACM Press.
- Grimme, C., Lepping, J., & Papaspyrou, A. (2007). Exploring the Behavior of Building Blocks for Multi-Objective Variation Operator Design using Predator-Prey Dynamics. In Thierens, D. et al. (Hrsg.), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, (S. 805–812)., London. ACM Press.

- Grimme, C., Lepping, J., & Papaspyrou, A. (2008a). Discovering Performance Bounds for Grid Scheduling by using Evolutionary Multiobjective Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, (S. 1491–1498)., New York, NY, USA. ACM.
- Grimme, C., Lepping, J., & Papaspyrou, A. (2008b). The Parallel Predator-Prey Model: A Step Towards Practical Application. In Rudolph, G. et al. (Hrsg.), *Proceedings of the 10th International Conference on Parallel Problem Solving From Nature (PPSN X)*, Band 5199 von *Lecture Notes in Computer Science*, (S. 681–690). Springer.
- Grimme, C., Lepping, J., & Papaspyrou, A. (2009). Adapting to the Habitat: On the Integration of Local Search into the Predator-Prey Model. In Ehrgott, M., Fonseca, C. M., Gandibleux, X., Hao, J.-K., & Sevaux, M. (Hrsg.), *Proceedings of the Fifth International Conference on Evolutionary Multi-Criterion Optimization (EMO)*, Band 5467 von *Lecture Notes in Computer Science*, (S. 510–524).
- Grimme, C., Lepping, J., & Papaspyrou, A. (2011). Parallel Predator-Prey Interaction for Evolutionary Multi-Objective Optimization. *Natural Computing*, 1–15 (online).
- Grimme, C., Lepping, J., & Schwiegelshohn, U. (2011). Multi-criteria scheduling: an agent-based approach for expert knowledge integration. *Journal of Scheduling*, 1–15 (online).
- Grimme, C. & Schmitt, K. (2006). Inside a Predator-Prey Model for Multi-Objective Optimization: A Second Study. In Beyer, H.-G. et al. (Hrsg.), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, (S. 707–714)., New York. ACM Press.
- Gwiazda, T. D. (2006). *Genetic Algorithms Reference (Vol.1): Crossover for single-objective numerical optimization problems*. Lomianki: Tomasz Gwiazda.
- Haimes, Y. Y., Lasdon, L. S., & Wismer, D. A. (1971). On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transaction on Systems, Man, and Cybernetics*, 1, 296–297.
- Harada, K., Sakuma, J., & Kobayashi, S. (2006). Local Search for Multiobjective Function Optimization: Pareto Descent Method. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, (S. 659–666)., New York, NY, USA. ACM.
- Haupt, R. (1989). A Survey of Priority Rule-Based Scheduling. *OR Spectrum*, 11(1), 3–16.
- Hennig, W. (1995). *Genetik* (1. Aufl.). Springer.

- Hillis, W. D. (1990). Co-evolving parasites improve simulated evolution as an optimization procedure. *Phys. D*, 42, 228–234.
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems* (1. Aufl.). MIT Press.
- Hoogeveen, H. (2005). Multicriteria scheduling. *European Journal of Operational Research*, 167(3), 592–623.
- Horn, J., Nafpliotis, N., & Goldberg, D. E. (1994). A Niche Pareto Genetic Algorithm for Multiobjective Optimization. In *International Conference on Evolutionary Computation*, (S. 82–87).
- Jackson, J. R. (1955). Scheduling a Production Line to Minimize Maximum Tardiness. Management Science Research Project, Research Report 43, University of California, Los Angeles.
- John, F. (1948). Extremum problems with inequalities as subsidiary conditions. In *Studies and Essays, Courant Anniversary Volume* (S. 187–204). New York: Interscience.
- Judt, L., Mersmann, O., & Naujoks, B. (2012). Effect of SMS-EMOA Parameterizations on Hypervolume Decreases. In *Proceedings of the Learning and Intelligent Optimization Conference*, Paris, France. Springer.
- Kawaguchi, T. & Kyan, S. (1986). Worst Case Bound of an LRF Schedule for the Mean Weighted Flow-Time Problem. *SIAM Journal on Computing*, 15(4), 1119–1129.
- Kennedy, J. & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings., IEEE International Conference on Neural Networks, 1995*, Band 4, (S. 1942–1948). IEEE Press.
- Knowles, J. & Corne, D. (2000). Approximating the Nondominated Front Using the Pareto Archived Evolution Strategies. *Evolutionary Computation*, 8(2), 149–172.
- Knowles, J. & Corne, D. (2007). Quantifying the effects of objective space dimension in evolutionary multiobjective optimization. In *Proceedings of the 4th international conference on Evolutionary multi-criterion optimization*, Band 4403 von *Lecture Notes in Computer Science*, (S. 757–771)., Berlin, Heidelberg. Springer.
- Knuth, D. E. (1997). *The art of computer programming, volume 3: sorting and searching* (3. Aufl.). Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

- Kuhn, H. W. & Tucker, A. W. (1951). Nonlinear Programming. In *Proceedings on the Second Berkeley Symposium on Mathematical Statistics and Probability*, (S. 481–492). University of California Press.
- Kursawe, F. (1999). *Grundlegende empirische Untersuchungen der Parameter von Evolutionsstrategien – Metastrategien*. Dissertation Technische Universität Dortmund, Dortmund.
- Laumanns, M. (1999). Ein verteiltes Räuber-Beute-Modell zur mehrkriteriellen Optimierung. Diplomarbeit, Universität Dortmund, Fachbereich Informatik, Lehrstuhl für Systemanalyse.
- Laumanns, M., Rudolph, G., & Schwefel, H.-P. (1998). A Spatial Predator-Prey Approach to Multi-Objective Optimization: A Preliminary Study. In T. Bäck, A. E. Eiben, M. Schoenauer, & H.-P. Schwefel (Hrsg.), *Parallel Problem Solving From Nature V* (S. 241–249). Berlin: Springer.
- Lee, C. Y. (1994). Genetic algorithms for single machine job scheduling with common due date and symmetric penalties. *Journal of the Operations Research Society of Japan*, 37(2), 83–95.
- Lenstra, J., Kan, A. R., & Brucker, P. (1977). Complexity of Machine Scheduling Problems. In B. K. P.L. Hammer, E.L. Johnson & G. Nemhauser (Hrsg.), *Studies in Integer Programming*, Band 1 von *Annals of Discrete Mathematics* (S. 343–362). Elsevier.
- Li, X. (2003). A Real-Coded Predator-Prey Genetic Algorithm for Multiobjective Optimization. In *Evolutionary Multi-Criterion Optimization Conference*, (S. 207–221).
- Luque, G., Alba, E., & Dorronsoro, B. (2005). Parallel Genetic Algorithms. In E. Alba (Hrsg.), *Parallel Metaheuristics: A New Class of Algorithms* (S. 107–125). John Wiley & Sons.
- Mendel, G. (1865). Versuche über Pflanzenhybriden. *Verhandlungen des naturforschenden Vereines in Brünn*, 4, 3–47.
- Michalewicz, Z. (1999). *Genetic Algorithms + Data Structures = Evolution Programs* (3. Aufl.). Berlin: Springer.
- Miettinen, K. (1999). *Nonlinear Multiobjective Optimization*. Kluwer's International Series in Operations Research & Management Science. Boston: Kluwer Academic Publishers.
- Moore, J. M. (1968). An  $n$  Job, One Machine Sequencing Algorithm for Minimizing the Number of Late Jobs. *Management Science*, 15, 102–109.

- Moscato, P., Cotta, C., & Mendes, A. (2004). Memetic algorithms. In G. Onwubolu & B. Babu (Hrsg.), *New Optimization Techniques in Engineering* (S. 53–85). Springer-Verlag.
- Motwani, R. & Raghavan, P. (1995). *Randomized Algorithms*. Cambridge University Press.
- Nachtigall, W. (2002). *Bionik, Grundlagen und Beispiele für Ingenieure und Naturwissenschaftler* (2. Aufl.). Springer.
- Okuda, T., Hiroyasu, T., Miki, M., & Watanabe, S. (2002). DCMOGA: Distributed Cooperation Model of Multi-Objective Genetic Algorithm. In *Proceedings of the PPSN/SAB Workshop on Multiobjective Problem Solving from Nature II (MPSN-II)*.
- Pareto, V. (1971). *Manuale d'economia politica (engl. von Schwier, Ann S. & Page, Alfred N.)*. London: Macmillan.
- Parsopoulos, K. E., Tasoulis, D. K., & Vrahatis, M. N. (2004). Multiobjective Optimization Using Parallel Vector Evaluated Particle Swarm Optimization. In *In Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA 2004)*, (S. 823–828). ACTA Press.
- Parsopoulos, K. E. & Vrahatis, M. N. (2002). Particle swarm optimization method in multiobjective problems. In *Proceedings of the 2002 ACM Symposium on Applied computing*, (S. 603–607)., New York, NY, USA. ACM.
- Peschel, M. (1980). *Ingenieurtechnische Entscheidungen. Modellbildung und Steuerung mit Hilfe der Polyoptimierung*. Berlin: VEB Verlag Technik.
- Pinedo, M. (2008). *Scheduling - Theory, Algorithms, and Systems* (3. Aufl.). Springer.
- Popovici, E., Bucci, A., Wiegand, R. P., & de Jong, E. D. (2011). Coevolutionary Principles. In G. Rozenberg, T. Bäck, & J. N. Kok (Hrsg.), *Handbook of Natural Computing*. Springer.
- Powell, D. & Hollingsworth, J. (2007). A NSGA-II, web-enabled, parallel optimization framework for NLP and MINLP. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, (S. 2145–2150)., New York, NY, USA. ACM.
- Rechenberg, I. (1971). *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Dissertation Technische Universität Berlin, Berlin.
- Rechenberg, I. (1994). *Evolutionsstrategie '94*. Stuttgart: Frommann-Holzboog.

- Reyes-Sierra, M. & Coello Coello, C. A. (2006). Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art. *International Journal of Computational Intelligence Research*, 2(3), 287–308.
- Ronald, S. (1997). Robust Encodings in Genetic Algorithms: A Survey of Encoding Issues. In *Evolutionary Computation, 1997., IEEE International Conference on*, (S. 43–48). IEEE Press.
- Rothlauf, F. (2006). *Representations for Genetic and Evolutionary Algorithms*. Heidelberg: Springer.
- Rowe, J., Vinsen, K., & Marvin, N. (1996). Parallel GAs for Multiobjective Functions. In Alander, J. T. (Hrsg.), *Second Nordic Workshop on Genetic Algorithms and Their Applications (2NWGA)*, (S. 61–70)., Vaasa, Finland. University of Vaasa.
- Rudenko, O. & Schoenauer, M. (2004). Dominance Based Crossover Operator for Evolutionary Multi-objective Algorithms. In Yao, X. et al. (Hrsg.), *Parallel Problem Solving from Nature VIII*, Band 3242 von *Lecture Notes in Computer Science*, (S. 812–821). Springer.
- Rudolph, G. & Schwefel, H.-P. (2008). Simulated evolution under multiple criteria conditions revisited. In *Proceedings of the 2008 IEEE World Conference on Computational intelligence: Research Frontiers*, (S. 249–261)., Berlin, Heidelberg. Springer.
- Runarsson, T. P. (2003). An Asynchronous Parallel Evolution Strategy. *Int. J. Journal of Computational Intelligence and Applications*, 3(4), 1–14.
- Schaffer, J. D. (1985). Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms*, (S. 93–100)., Hillsdale, NJ, USA. L. Erlbaum Associates Inc.
- Schaffer, J. D. & Grefenstette, J. J. (1985). Multi-Objective Learning via Genetic Algorithms. In *International Joint Conference Artificial Intelligence*, (S. 593–595).
- Schmitt, K., Mehnen, J., & Michelitsch, T. (2005). Using predators and preys in evolution strategies. In Beyer, H.-G. et al. (Hrsg.), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, Band 1, (S. 827–828)., New York. ACM Press.
- Schuetze, O., Sanchez, G., & Coello Coello, C. A. (2008). A New Memetic Strategy for the Numerical Treatment of Multi-Objective Optimization Problems. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, (S. 705–712)., Atlanta, GA, USA. ACM.

- Schwefel, H.-P. (1968). Experimentelle Optimierung einer Zweiphasendüse, Teil I. Bericht Nr. 35 zum Projekt MHD–Staustrahlrohr 11.034/68, AEG Forschungsinstitut, Berlin.
- Schwefel, H.-P. (1975). *Evolutionsstrategie und numerische Optimierung*. Dr.-Ing. Dissertation, Technische Universität Berlin, Fachbereich Verfahrenstechnik.
- Schwefel, H.-P. (1995). *Evolution and Optimum Seeking*. New York: Wiley.
- Schwefel, H.-P. & Kursawe, F. (1998). On natural life's tricks to survive and evolve. In *Proceedings of the IEEE International Conference on Evolutionary Computation*, (S. 1–8).
- Schwefel, H.-P. & Rudolph, G. (1995). Contemporary evolution strategies. In Morán, F., Moreno, A., Merelo, J. J., & Chacón, P. (Hrsg.), *Advances in Artificial Life – Third European Conference on Artificial Life*, (S. 893–907)., Berlin. Springer.
- Schwiegelshohn, U. (2011). An alternative proof of the Kawaguchi-Kyan bound for the Largest-Ratio-First rule. *Operations Research Letters*, 39(4), 255–259.
- Scriven, I., Ireland, D., Lewis, A., Mostaghim, S., & Branke, J. (2008). Asynchronous multiple objective particle swarm optimisation in unreliable distributed environments. In *Proceedings of the International Conference on Evolutionary Computation*, (S. 2481–2486). IEEE Press.
- Seshadri, A. (2009). NSGA - II: A multi-objective optimization algorithm. MATLAB Central, Implementierung.
- Shukla, P. K. (2007). On Gradient Based Local Search Methods in Unconstrained Evolutionary Multi-objective Optimization. In *Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization*, Band 4403 von *Lecture Notes in Computer Science*, (S. 96–110)., Heidelberg. Springer.
- Siirola, J. D., Huan, S., & Westerberg, A. W. (2004). Computing Pareto fronts using distributed agents. *Computers & Chemical Engineering*, 29(1), 113–126.
- Simon, C. P. (1976). Scalar and vector maximization: Calculus techniques with economics applications. Technischer Bericht Discussion Paper No. 257, University of Michigan.
- Smith, W. E. (1956). Various Optimizers for Single-stage Production. *Naval Research Logistics Quarterly*, 3, 59–66.
- Spall, J. C. (1992). Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation. *IEEE Transactions on Automatic Control*, 37, 332–341.

- Spall, J. C. (1998). An Overview of the Simultaneous Perturbation Method for Efficient Optimization. *Johns Hopkins APL Technical Digest*, 19(4), 482–492.
- Srinivas, N. & Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3), 221–248.
- Stacey, A., Jancic, M., & Grundy, I. (2003). Particle swarm optimization with mutation. In *Proceedings of the Congress on Evolutionary Computation, 2003.*, Band 2, (S. 1425–1430). IEEE-Press.
- Storch, U. & Wiebe, H. (1999). *Lehrbuch der Mathematik* (2. Aufl.), Band 2. Heidelberg: Spektrum.
- Süer, G. A., Báez, E., & Czajkiewicz, Z. (1993). Minimizing the number of tardy jobs in identical machine scheduling. *Computers and Industrial Engineering*, 25(1–4), 243–246.
- Talbi, E.-G., Mostaghim, S., Okabe, T., Ishibuchi, H., Rudolph, G., & Coello Coello, C. A. (2008). Parallel Approaches for Multiobjective Optimization. In *Multiobjective Optimization: Interactive and Evolutionary Approaches* (S. 349–372). Berlin, Heidelberg: Springer-Verlag.
- Thomsen, R., Rickers, P., & Krink, T. (2000). A Religion-Based Spatial Model for Evolutionary Algorithms. In *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature, PPSN VI*, (S. 817–826)., London, UK. Springer-Verlag.
- T'kindt, V. & Billaut, J.-C. (2006). *Multicriteria Scheduling. Theory, Models and Algorithms* (2. Aufl.). Springer.
- Tomassini, M. (2005). *Spatially Structured Evolutionary Algorithms: Artificial Evolution in Space and Time*. Natural Computing Series. Springer.
- Ursem, R. (1999). Multinational evolutionary algorithms. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, Band 3, (S. 1633–1640).
- van Wassenhove, L. N. & Gelders, F. (1980). Solving a Bicriterion Scheduling Problem. *European Journal of Operational Research*, 2(4), 281–290.
- Veldhuizen, D. V. & Lamont, G. (1998). Multiobjective Evolutionary Algorithm Research: A History and Analysis. Technical Report Dept. Elec. Comput. Eng. Air Force TR-98-03, Air Force Inst. Technol.
- Xiong, S. & Li, F. (2003). Parallel strength Pareto multi-objective evolutionary algorithm for optimization problems. In *Proceedings of the Congress on Evolutionary Computation*, Band 4, (S. 2712–2718).

- Zitzler, E. (1999). *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. Dissertation Swiss Federal Institute of Technology (ETH), Zürich.
- Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2), 173–195.
- Zitzler, E. & Künzli, S. (2004). Indicator-Based Selection in Multiobjective Search. In Yao, X. et al. (Hrsg.), *Conference on Parallel Problem Solving from Nature (PPSN VIII)*, Band 3242 von *Lecture Notes in Computer Science*, (S. 832–842). Springer.
- Zitzler, E., Laumanns, M., & Thiele, L. (2002). SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In Giannakoglou, K. et al. (Hrsg.), *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*, (S. 95–100). International Center for Numerical Methods in Engineering (CIMNE).
- Zitzler, E. & Thiele, L. (1999). Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4), 257—271.
- Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2), 117–132.
- Zuckerman, D. (1992). A Technique for Lower Bounding the Cover Time. In *SIAM Journal on Discrete Mathematics*, (S. 81–87). ACM Press.