

---

# MASTERARBEIT

---

Dipl.-Math.(FH)  
**Steffen Kux**

**Hybride Optimierungsstrategien für  
komplexe technische  
Aufgabenstellungen**

2011



Fakultät: **Mathematik/Naturwissenschaften/Informatik**

---

# **MASTERARBEIT**

---

## **Hybride Optimierungsstrategien für komplexe technische Aufgabenstellungen**

Autor:

**Dipl.-Math.(FH) Steffen Kux**

Studiengang:

Diskrete und Computerorientierte Mathematik

Seminargruppe:

ZD09w1

Erstprüfer:

Prof. Dr. rer. nat. Thomas Villmann

Zweitprüfer:

Dr. rer. nat. Horst Liebermann

Mittweida, Oktober 2011



## **Bibliographische Angaben**

Kux, Steffen: Hybride Optimierungsstrategien für komplexe technische Aufgabenstellungen, 152 Seiten mit Anhang, 58 Abbildungen, 3 Tabellen, 14 Algorithmen, Hochschule Mittweida(FH), Fakultät Mathematik/Naturwissenschaften/Informatik

Masterarbeit, 2011

## **Referat**

Im Entwicklungsprozess komplexer, technischer Systeme spielen Simulationsmethoden, denen parametrische Modelle zugrunde liegen, eine wichtige Rolle. Diese Modelle bilden physikalische oder technische Eigenschaften von Bauteilen oder ganzen Funktionssystemen ab und ermöglichen deren Simulation und virtuelle Entwicklung. Durch den Einsatz geeigneter mathematischer Optimierungsmethoden ist es möglich, das Potential dieser Systeme besser auszuschöpfen und wichtige Teile dieses Prozesses zu automatisieren.

Die Kombination verschiedener Verfahren ist einer der erfolgversprechendsten Trends in der Optimierung. Dabei spielt insbesondere die Hybridisierung naturinspirierter, heuristischer Verfahren eine wichtige Rolle. Mit hybriden Optimierungsstrategien wird beispielsweise das Ziel verfolgt, die guten Eigenschaften verschiedener Verfahren zu vereinen, vorhandene Informationen besser auszunutzen oder Verfahrensparameter der Optimierungsverfahren adaptiv im Verlauf einer Optimierung geeignet anzupassen.

Im Rahmen dieser Arbeit werden verschiedene hybride Strategien untersucht und bewertet. Als Ergebnis wird das neue hybride Optimierungsverfahren AHMOS (Adaptive Hybride Multi-criterion Optimization Strategy) konstruiert.



# Vorwort

Diese Masterarbeit entstand im Rahmen meines Masterstudiums im Studiengang „Diskrete und Computerorientierte Mathematik“ an der Hochschule Mittweida, welches ich fast zehn Jahre nach Abschluss meines Diploms in „Angewandter Mathematik“ an der gleichen Hochschule aufgenommen habe. Mein Dank gilt den Professorinnen und Professoren der Fachgruppe Mathematik in der Fakultät Mathematik/Naturwissenschaften/Informatik der Hochschule Mittweida für Ihre Unterstützung und individuelle Betreuung in dieser Zeit. Mein besonderer Dank gilt an dieser Stelle Frau Professor Regina Fischer, auf deren Motivation hin ich mich überhaupt erst entschlossen hatte, dieses (weitere) Studium parallel zu meiner Arbeit zu beginnen.

In den vergangenen acht Jahren hatte ich mich im Rahmen meiner Arbeit bei der IAV GmbH intensiv mit Methoden der Optimierung in Verbindung mit aufwändigen technischen Simulationen beschäftigt. Viele Ideen, die im Rahmen dieser Arbeit näher untersucht und umgesetzt wurden, basieren auf den Erfahrungen, die sich beim Lösen dieser Problemstellungen angesammelt haben. Deshalb war es mir ein besonderes Anliegen, diese Erkenntnisse zu vertiefen und in Form eines neu konzipierten Optimierungsframeworks, der im Rahmen dieser Arbeit entstanden ist, zu implementieren und zur Anwendung zu bringen.

Mein besonderer Dank gilt auch Martin Stöcker für die vielen anregenden Fachdiskussionen und Brainstormings zum Thema Optimierung und wertvollen Tipps.

Ich danke auch Dr. Horst Liebermann, der mich von der betrieblichen Seite aus betreute und viele wertvolle Anregung zur Gestaltung der Arbeit gab.

Nicht unerwähnt lassen möchte ich an dieser Stelle meine Frau Michaela und meine vier Kinder Elisabeth, Antonina, Kilian und Janinah, denen in der Zeit dieses Studiums und besonders in der Zeit der Bearbeitung dieser Masterarbeit so manche zeitliche Einschränkung abverlangt wurde, die mich aber jederzeit unterstützt und motiviert haben. Ihnen gilt mein ganz persönlicher Dank. Weiterhin möchte ich auch Steeve Meyner für seine Ideen bei der textlichen Gestaltung der Arbeit recht herzlich danken.

Diese Arbeit wurde mit LyX geschrieben und mit  $\text{\LaTeX}$  formatiert.





# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>1</b>
1.1. Motivation . . . . .	1
1.2. Gliederung . . . . .	4
1.3. Zielstellung . . . . .	5
<b>2. Nichtlineare Multikriterielle Optimierung</b>	<b>7</b>
2.1. Grundlagen . . . . .	7
2.2. Herausforderungen bei der Optimierung von technischen Problemstellungen . .	10
2.2.1. Aufwändige, restringierte Blackbox-Probleme in der Technik . . . . .	10
2.2.1.1. Mehrere widersprüchliche Ziele . . . . .	11
2.2.1.2. Unlösbare Varianten . . . . .	12
2.2.1.3. Viele lokale Extrema . . . . .	13
2.2.1.4. Schmale und nichtzusammenhängende Restriktionsbereiche .	13
2.2.2. Parallelisierung . . . . .	14
<b>3. Ausgewählte Optimierungsverfahren</b>	<b>15</b>
3.1. Überblick . . . . .	15
3.2. Deterministische Verfahren und deren multikriterielle Erweiterungen . . . . .	17
3.2.1. Gradientenverfahren . . . . .	17
3.2.2. Gradientenfreie Verfahren . . . . .	18
3.2.2.1. NELDER-MEAD-Simplex-Verfahren . . . . .	18
3.2.2.2. DiRect-Verfahren . . . . .	21
3.2.3. Multikriterielle Optimierung durch Überführung in einkriterielle Ersatz-	
optimierungsaufgaben . . . . .	23
3.2.3.1. Gewichtete-Summen-Methode . . . . .	24
3.2.3.2. $\epsilon$ -Constraint-Methode . . . . .	25
3.2.3.3. Gewichtete-Metrik-Methode . . . . .	26
3.2.3.4. Nutzenfunktionsmethode . . . . .	26
3.2.3.5. PARETO-Scan . . . . .	27
3.2.3.6. Zusammenfassung . . . . .	27
3.3. Stochastische Verfahren . . . . .	28
3.3.1. Zufallszahlen . . . . .	28
3.3.2. Monte-Carlo-Methode . . . . .	29

3.3.3.	Naturinspirierte Verfahren . . . . .	31
3.3.3.1.	Evolutionäre Verfahren . . . . .	32
3.3.3.2.	Partikelschwarm Verfahren . . . . .	35
3.3.3.3.	Ameisen Verfahren . . . . .	38
3.3.3.4.	Simulated Annealing . . . . .	40
3.3.4.	Adaptive Ersatzmodell-Verfahren . . . . .	42
3.4.	Bewertung der Verfahren bezüglich der Herausforderungen bei der Optimierung technischer Probleme . . . . .	43
<b>4.</b>	<b>Hybride Strategien</b>	<b>45</b>
4.1.	Überblick . . . . .	45
4.2.	Globale Suche - Lokale Verfeinerung . . . . .	47
4.2.1.	Kriterien zur Umschaltung der Suchstrategie . . . . .	49
4.2.2.	Multikriterielle Erweiterung . . . . .	53
4.2.2.1.	Multikriterielle Umschaltkriterien . . . . .	53
4.2.2.2.	Auswahl der Ausgangspunkte für die lokale Suche . . . . .	54
4.2.2.3.	Suche mit eingeschränkten Wertebereichen der Optimierungsvariablen . . . . .	55
4.3.	Ersatzmodellssuche . . . . .	55
4.3.1.	Approximationsverfahren . . . . .	56
4.3.1.1.	Multiple Regression . . . . .	56
4.3.1.2.	Moving-Least-Squares-Verfahren . . . . .	57
4.3.2.	Optimierung mit Ersatzmodell und Nachrechnung . . . . .	58
4.3.2.1.	Ersatzmodellssuche während der Optimierung . . . . .	58
4.3.2.2.	Ersatzmodellssuche als Nachiteration . . . . .	60
4.4.	Bewertung der Strategien bezüglich der Herausforderungen bei der Optimierung technischer Probleme . . . . .	60
4.5.	Das hybride Optimierungsverfahren AHMOS . . . . .	61
4.5.1.	Überblick und Struktur . . . . .	62
4.5.2.	Parallelisierung . . . . .	64
4.5.3.	Ausblick . . . . .	65
<b>5.</b>	<b>Benchmark der Optimierungsverfahren</b>	<b>67</b>
5.1.	Test-Aufgaben für einkriterielle Optimierungsaufgaben . . . . .	68
5.1.1.	Testfunktion: RASTRIGIN . . . . .	68
5.1.1.1.	Definition . . . . .	68
5.1.1.2.	Zielstellung . . . . .	69
5.1.1.3.	Test und Auswertung . . . . .	69
5.1.2.	Testfunktion: ROSENBROCK . . . . .	74
5.1.2.1.	Definition . . . . .	74

5.1.2.2.	Zielstellung . . . . .	74
5.1.2.3.	Test und Auswertung . . . . .	74
5.1.3.	Testfunktion: HIMMELBLAU . . . . .	78
5.1.3.1.	Definition . . . . .	78
5.1.3.2.	Zielstellung . . . . .	78
5.1.3.3.	Test und Auswertung . . . . .	78
5.1.4.	Testfunktion: DEB2 . . . . .	81
5.1.4.1.	Definition . . . . .	81
5.1.4.2.	Zielstellung . . . . .	81
5.1.4.3.	Test und Auswertung . . . . .	81
5.1.5.	Testfunktion: ALPENTAL . . . . .	84
5.1.5.1.	Definition . . . . .	84
5.1.5.2.	Zielstellung . . . . .	86
5.1.5.3.	Test und Auswertung . . . . .	86
5.1.6.	Testfunktion: MINER-S-1 . . . . .	87
5.1.6.1.	Definition . . . . .	87
5.1.6.2.	Zielstellung . . . . .	88
5.1.6.3.	Test und Auswertung . . . . .	88
5.2.	Test-Aufgaben für multikriterielle Optimierungsaufgaben . . . . .	92
5.2.1.	Testfunktion: DTLZ1 . . . . .	94
5.2.1.1.	Definition . . . . .	94
5.2.1.2.	Zielstellung . . . . .	95
5.2.1.3.	Test und Auswertung . . . . .	96
5.2.2.	Testfunktion: DTLZ2 . . . . .	97
5.2.2.1.	Definition . . . . .	97
5.2.2.2.	Zielstellung . . . . .	98
5.2.2.3.	Test und Auswertung . . . . .	98
5.2.3.	Testfunktion: MINER-M-1 . . . . .	101
5.2.3.1.	Definition . . . . .	101
5.2.3.2.	Zielstellung . . . . .	101
5.2.3.3.	Test und Auswertung . . . . .	102
5.3.	Zusammenfassung und Schlussfolgerung . . . . .	103
<b>6.</b>	<b>Zusammenfassung und Ausblick</b>	<b>107</b>
<b>A.</b>	<b>Legendre-Polynome</b>	<b>121</b>
<b>B.</b>	<b>Erläuterung: Verteilungsplot</b>	<b>123</b>

<b>C. Parametrisierung der Optimierungsverfahren für den Vergleichstest</b>	<b>125</b>
C.1. MCO . . . . .	125
C.2. NSGA-II . . . . .	125
C.3. Nelder-Mead . . . . .	126
C.4. AHMOS . . . . .	126
C.4.1. Globale Parameter . . . . .	126
C.4.2. Globale Suche mit NSGA-II . . . . .	126
C.4.3. Lokale Suche mit NELDER-MEAD . . . . .	126

# Abbildungsverzeichnis

2.1.	Beispiel mit 2 Variablen und 2 Zielen (Minimierung): zulässige Menge $S$ , Ergebnismenge $F_S$ , PARETO-Menge $X$ , PARETO-Front $P$ . . . . .	9
3.1.	Übersicht Nichtlineare Optimierungsverfahren . . . . .	16
3.2.	NELDER-MEAD-Operationen an einem Beispiel mit 2 Variablen (A) ... Reflektion (B) ... Expansion (C) ... Kontraktion (D) ... Schrumpfen . . . . .	20
3.3.	untere Schranken einer LIPSCHITZ-stetigen Funktion . . . . .	21
3.4.	untere Schranken des DiRect-Verfahrens . . . . .	22
3.5.	Beispiel zum DiRect-Verfahren: mögliche Teilungen des zulässigen Bereichs einer Funktion mit 2 Variablen (Zwischenschritte ausgelassen) . . . . .	23
3.6.	Gewichtete-Summen-Methode (Beispiel mit 2 Zielen) A ... erreichbarer Punkt $\rightarrow$ auf konvexem Bereich der PARETO-Menge B ... nicht eindeutig erreichbarer Punkt $\rightarrow$ am Rand des konvexen Bereichs der PARETO-Menge C ... nicht erreichbarer Punkt $\rightarrow$ nicht auf konvexem Bereich der PARETO-Menge . . . . .	24
3.7.	$\varepsilon$ -Constraint-Methode (Beispiel mit 2 Zielen) $\varepsilon_A$ ... keine Einschränkung des Zielgebietes $\varepsilon_B$ ... Einschränkung zur Wahl des Punktes B $\varepsilon_C$ ... die zulässige Menge ist leer $\rightarrow$ keine Lösung . . . . .	25
3.8.	Bewegung eines Partikels . . . . .	37
3.9.	schematische Bewegung eines Partikels bei MOA . . . . .	37
3.10.	Prinzip der Futtersuche einer Ameisenkolonie (1) ... Ameisen suchen zufällig im Zielgebiet (2) ... Erfolgreich zurückgekehrte Ameisen locken andere auf ihre Spur (3) ... die meisten Ameisen sind auf dem kürzesten Weg zwischen Hügel und Futterquelle . . . . .	38
3.11.	Kristallgitter (verspanntes und regelmäßiges) . . . . .	40
4.1.	Verteilung der Individuen einer Population (Beispiel) links ... nicht homogen, die Individuen sind im ganzen Lösungsraum verteilt rechts ... homogen, die Individuen konzentrieren sich in einem Bereich . . . . .	50
4.2.	Regressionsanalyse der Individuen einer Population (Beispiel) links ... Regression fittet den betrachteten Bereich des Zielraumes schlecht rechts ... Regression fittet den betrachteten Bereich des Zielraumes gut . . . . .	51
4.3.	Fortschritt der Optimierung (Beispiel) . . . . .	52

## Abbildungsverzeichnis

4.4. Fortschritt einer multikriteriellen Optimierung (Beispiel mit 2 ZF)	
links ... die aktuelle (neue) Population dominiert die Vergleichs- (alte) Population	
rechts ... die aktuelle (neue) Population dominiert die Vergleichs- (alte) Population nicht . . . . .	54
4.5. Struktur des hybriden Optimierungsverfahrens AHMOS . . . . .	63
5.1. Testfunktion: 2-dimensionale RASTRIGIN . . . . .	70
5.2. Auswertung: RASTRIGIN mit 2 Variablen . . . . .	71
5.3. Auswertung (bestes Verfahren): RASTRIGIN mit 2 Variablen . . . . .	71
5.4. Auswertung: RASTRIGIN mit 5 Variablen . . . . .	72
5.5. Auswertung (bestes Verfahren): RASTRIGIN mit 5 Variablen . . . . .	72
5.6. Auswertung: RASTRIGIN mit 9 Variablen . . . . .	73
5.7. Auswertung (bestes Verfahren): RASTRIGIN mit 9 Variablen . . . . .	73
5.8. Testfunktion: 2-dimensionale ROSENBROCK . . . . .	75
5.9. Auswertung: ROSENBROCK mit 3 Variablen . . . . .	76
5.10. Auswertung (bestes Verfahren): ROSENBROCK mit 3 Variablen . . . . .	76
5.11. Auswertung: ROSENBROCK mit 5 Variablen . . . . .	77
5.12. Auswertung (bestes Verfahren): ROSENBROCK mit 5 Variablen . . . . .	77
5.13. Testfunktion: HIMMELBLAU . . . . .	79
5.14. Auswertung: HIMMELBLAU (2 Variablen) . . . . .	80
5.15. Auswertung (bestes Verfahren): HIMMELBLAU (2 Variablen) . . . . .	80
5.16. Testfunktion: DEB2 (zulässiger Bereich hervorgehoben) . . . . .	82
5.17. Auswertung: DEB2 (2 Variablen) . . . . .	83
5.18. Auswertung (bestes Verfahren): DEB2 (2 Variablen) . . . . .	83
5.19. Testfunktion: ALPENTAL (Unzulässiger Bereich grau hervorgehoben) . . . . .	85
5.20. Auswertung: ALPENTAL (2 Variablen) . . . . .	86
5.21. Auswertung (bestes Verfahren): ALPENTAL (2 Variablen) . . . . .	87
5.22. Testfunktion: 2-dimensionale MINER-S-1 . . . . .	89
5.23. Auswertung: MINER-S-1 mit 2 Variablen . . . . .	90
5.24. Auswertung (bestes Verfahren): MINER-S-1 mit 2 Variablen . . . . .	90
5.25. Auswertung: MINER-S-1 mit 5 Variablen . . . . .	91
5.26. Auswertung (bestes Verfahren): MINER-S-1 mit 5 Variablen . . . . .	91
5.27. Testfunktion: DTLZ1 mit 2 Zielen und 2 Variablen . . . . .	95
5.28. Auswertung: DTLZ1 mit 2 Zielen und 6 Variablen . . . . .	96
5.29. Auswertung (bestes Verfahren): DTLZ1 (2 Ziele und 6 Variablen) . . . . .	97
5.30. Testfunktion: DTLZ2 mit 2 Zielen und 2 Variablen . . . . .	98
5.31. Auswertung: DTLZ2 mit 2 Zielen und 5 Variablen . . . . .	99
5.32. Auswertung (bestes Verfahren): DTLZ2 (2 Ziele und 5 Variablen) . . . . .	99
5.33. Auswertung: DTLZ2 mit 4 Zielen und 7 Variablen . . . . .	100
5.34. Auswertung (bestes Verfahren): DTLZ2 (4 Ziele und 7 Variablen) . . . . .	100

5.35. Testfunktion: MINER-M-1 mit 2 Zielen und 2 Variablen, ohne Restriktionsbereiche (siehe Abbildung 5.36) . . . . .	102
5.36. Restriktionen der Testfunktion: MINER-M-1 mit 2 Zielen und 2 Variablen, grün: zulässiger Bereich . . . . .	103
5.37. Auswertung: MINER-M-1 mit 2 Zielen und 2 Variablen . . . . .	104
5.38. Auswertung (bestes Verfahren): MINER-M-1 (2 Ziele und 2 Variablen) . . . .	104
5.39. Auswertung: MINER-M-1 mit 5 Zielen und 5 Variablen . . . . .	105
5.40. Auswertung (bestes Verfahren): MINER-M-1 (5 Ziele und 5 Variablen) . . . .	105
B.1. Erläuterung Verteilungsplot . . . . .	123





# Algorithmenverzeichnis

3.1. Gradientenverfahren . . . . .	18
3.2. NELDER-MEAD-Verfahren . . . . .	20
3.3. DiRect-Verfahren . . . . .	23
3.4. Verfahren: Monte-Carlo-Optimierung . . . . .	30
3.5. Verfahren: Stochastic Design Approvement . . . . .	31
3.6. Evolutionäre Verfahren . . . . .	32
3.7. Partikelschwarm Verfahren . . . . .	36
3.8. Ameisenalgorithmus . . . . .	39
3.9. Simulated Annealing . . . . .	41
3.10. Adaptives Ersatzmodell-Verfahren . . . . .	42
4.1. Hybride Strategie: Globale Suche - Lokale Verfeinerung . . . . .	48
4.2. Hybride Strategie: Ersatzmodellssuche innerhalb der Optimierung . . . . .	59
4.3. Hybride Strategie: Ersatzmodellssuche als Nachiteration . . . . .	60
4.4. Hybrides Verfahren AHMOS . . . . .	64



# Tabellenverzeichnis

1.1. Beispiele komplexer technischer Optimierungsaufgaben aus dem Bereich der Motorenentwicklung . . . . .	3
3.1. Bewertung verschiedener Optimierungsverfahren . . . . .	44
4.1. Bewertung verschiedener hybriden Strategien . . . . .	61



# Abkürzungen

ACO	Ant Colony Optimization (engl.), Ameisenkolonieoptimierung
AHMOS	Adaptive Hybride Multikriterielle Optimierungsstrategie (engl. Adaptive Hybrid Multi-criterion Optimization Strategy)
CEM	Cross Entropy Methode
CFD	Computational Fluid Dynamics (engl.), Numerische Strömungsmechanik
CoV	Coefficient of Variation (engl.), Variationskoeffizient
EOA	Einkriterielle Optimierungsaufgabe
FEM	Finite Elemente Methode
GA	Genetischer Algorithmus
LHM	Latin Hypercube Methode
MCO	Monte Carlo Optimierung
MKS	Mehrkörpersimulation, engl. MBS (multi body simulation)
MOA	Multikriterielle Optimierungsaufgabe
MLS	Moving-Least-Squares
NLMOA	Nichtlineare Multikriterielle Optimierungsaufgabe
NLOA	Nichtlineare Optimierungsaufgabe
NSM	Nicht dominierte Sortierung Methode (engl. non-dominated sorting method)
OA	Optimierungsaufgabe
PSO	Partikelschwarm Optimierung
SA	Simulated Annealing
SDA	Stochastic Design Appovement (engl.) (Stochastische Design Verbesserung)
ZF	Zielfunktion



# Symbole

$CD(\mathbf{x}^{(j)})$	Crowding-Distance des Individuums $\mathbf{x}^{(j)}$
$CM(X_i)$	Konvergenzmetrik der PARETO-Menge $X_i$ des $i$ -ten Optimierungsverfahrens
$E(\mathbf{x})$	Energiezustand
$\epsilon$	Fehler
$\epsilon_{R^2}$	Schranke für Bestimmtheitsmaß für innere Ersatzmodellssuche
$\epsilon_{R^2}^{(ende)}$	Schranke für Bestimmtheitsmaß für Nachiteration
$f_m(\mathbf{x})$	Zielfunktion
$\hat{f}(\mathbf{x})$	Approximationsfunktion
$f_m^{max}$	maximaler Funktionswert der $m$ -ten Zielfunktion
$f_m^{min}$	minimaler Funktionswert der $m$ -ten Zielfunktion
$d_r^{(i)}$	relative Dominanz
$\frac{\partial}{\partial x_i} f(\mathbf{x})$	Partielle Ableitung der Funktion $f(\mathbf{x})$ nach $x_i$
$\nabla f(\mathbf{x})$	Gradient der Funktion $f(\mathbf{x})$
$F(\mathbf{x}^{(j)})$	Fitness des Individuums $\mathbf{x}^{(j)}$
$G$	Population
$g_j(\mathbf{x})$	Nebenbedingung
$\gamma_{kog}$	Einflussfaktor für kognitiven Anteil
$\gamma_{soz}$	Einflussfaktor für sozialen Anteil
$\gamma_{pareto}$	Einflussfaktor für PARETO-Anteil
$J$	Anzahl Nebenbedingungen
$k$	BOLTZMANN-Konstante
$K$	LIPSCHITZ-Konstante
$M$	Anzahl Zielfunktionen
$\mathbb{N}$	Menge der Natürlichen Zahlen

## Tabellenverzeichnis

$\mu$	Mittelwert
$N$	Anzahl Optimierungsvariablen
$N_{DOM}(\mathbf{x}^{(j)})$	Anzahl der Individuen, die $\mathbf{x}^{(j)}$ dominieren
$P$	PARETO-Front
$P_{BM}(\mathbf{x})$	BOLTZMANN-Verteilung
$\mathbf{p}^s$	Position des Partikels $s$
$\tilde{\mathbf{p}}^s$	bisher beste Position des Partikels $s$
$\tilde{\mathbf{p}}$	bisher beste Position des Schwarmes
$R(\mathbf{x}^{(j)})$	Rang des Elementes $\mathbf{x}^{(j)}$
$R^2$	Bestimmtheitsmaß
$\mathbb{R}$	Menge der Reellen Zahlen
$\mathbb{R}^+$	Menge der nichtnegativen Reellen Zahlen
$S$	zulässige Menge
$SI(\mathbf{x}^{(j)})$	Sharing Intensity des Individuums $\mathbf{x}^{(j)}$
$\mathbb{S}$	Simplex
$\mathcal{G}$	Schwarm
$\sigma$	Standardabweichung
$T$	Temperatur
$\theta_i$	minimaler Abstand des Elementes $i$ von Referenzmenge
$U(\dots)$	Nutzenfunktion
$\mathbf{v}^s$	Geschwindigkeit des Partikels $s$
$\mathbf{x}$	Optimierungsvariablen (Vektor), Individuum
$\mathbf{x}^*$	globales Optimum
$\mathbf{x}^{(j)}$	das $j$ -te Individuum
$X$	PARETO-Menge
$\mathbf{z}^*$	Idealpunkt



# 1

## Kapitel 1.

---

# Einleitung

## 1.1. Motivation

Im Entwicklungsprozess technischer Systeme spielen Simulationsmethoden, denen parametrische Modelle zugrunde liegen, eine wichtige Rolle. Diese Modelle bilden physikalische oder technische Eigenschaften von Bauteilen oder ganzen Funktionssystemen ab und ermöglichen deren Simulation und virtuelle Entwicklung. Durch den Einsatz geeigneter mathematischer Optimierungsmethoden ist es möglich, das Potential dieser Systeme besser auszuschöpfen und wichtige Teile dieses Prozesses zu automatisieren.

Aber auch trotz moderner Computertechnik können diese Simulationen oft sehr zeitaufwändig sein. Optimierungen, die zur Lösung komplexer technischer Problemstellungen eingesetzt werden, sind im Hinblick auf die nachfolgenden Betrachtungen durch folgende Eigenschaften charakterisiert:

**Nichtlineare Funktionen:** Zielfunktionen und Restriktionen werden durch nichtlineare Funktionen beschrieben. Häufig können diese Funktionen nicht geschlossen analytisch angegeben werden, da sie durch eine komplexe Simulation abgebildet werden.

**Aufwändige Funktionswertberechnung:** Die Funktionswertberechnung kann sehr aufwändig sein und deshalb kann in der real zur Verfügung stehenden Zeit nur eine stark begrenzte Anzahl von Berechnungen durchgeführt werden.

**Multikriterielle Aufgaben:** Für viele Anwendungen ist es erforderlich, dass mehrere widersprüchliche Ziele betrachtet werden. Das Ergebnis der Optimierung ist eine Menge möglicher Kompromisse, aus denen in der Auswertung eine *Beste Lösung* auszuwählen ist.

**Stark beschränkter Parameterraum:** Die Restriktionen beschränken den Parameterraum, so dass dieser schmal und zum Teil nicht zusammenhängend sein kann.

## 1. Einleitung

**Unlösbare Varianten:** Unlösbare Varianten können dadurch entstehen, dass verschiedene Parametrisierungen technisch nicht realisierbar sein können oder die Simulationssoftware<sup>1</sup> nicht in der Lage ist, für diese Parameterwerte die Berechnung auszuführen. Besonders zu Beginn einer Optimierung, wenn dem Optimierungsverfahren noch wenig Informationen zur Verfügung stehen und im gesamten Suchraum gesucht wird, können sehr viele unlösbare Varianten auftreten.

**Wenig Information über die Gestalt der Funktionen:** Realisiert zum Beispiel eine numerische Simulation die eigentliche Zielfunktion, so ist meistens nur wenig über deren exakte mathematische Gestalt bekannt (keine Gradienteninformationen, usw.).

**Viele lokale Extrema:** Besonders bei komplizierten, nichtlinearen Zielfunktionen und Nebenbedingungen können viele lokale Extrema auftreten. Die Herausforderung besteht dann darin, das globale Extremum zu finden.

In Tabelle 1.1 sind einige typische Beispiele von Optimierungsaufgaben aus der Antriebstrangentwicklung und deren Zuordnung zu den beschriebenen Eigenschaften dargestellt. Für die Lösung von Aufgaben, die eine oder mehrere dieser Eigenschaften besitzen, wurden verschiedene Optimierungsstrategien entwickelt<sup>2</sup>, die meist für bestimmte Probleme gut, für andere Probleme aber nicht so gut geeignet sind, so dass es kein Verfahren gibt, das für alle Problemstellungen gleichermaßen gut geeignet ist<sup>3</sup>.

Die vorliegende Arbeit beschäftigt sich mit Optimierungsverfahren, die zur Lösung derartiger Optimierungsaufgaben geeignet sind. Diese Verfahren sollten die folgenden Eigenschaften aufweisen:

**Mit relativ wenigen Funktionsauswertungen auskommen:** Da für den Einsatz der Optimierung in der Regel nur beschränkt viel Zeit zur Verfügung steht, können bei sehr lange dauernden Funktionsauswertungen (= lange dauernden Simulationen) auch nur wenige Berechnungen durchgeführt werden.

**Global arbeiten:** Verfahren, die zur Lösung von Problemstellungen verwendet werden, die viele lokale Extrema besitzen, sollen nicht in einem beliebigen lokalen Extrempunkt verharren, sondern die globale Lösung finden.

**Mit unlösbaren Varianten zurechtkommen:** Unlösbare Varianten, die z.B. aufgrund abgebrochener Simulationen entstehen, dürfen nicht zum Abbruch der Optimierung führen oder sie stark behindern.

---

<sup>1</sup>z.B. bedingt durch Softwarefehler oder Numerikprobleme

<sup>2</sup>siehe z.B. [58, 49, 52, 29, 34, 43, 46, 53, 2]

<sup>3</sup>auch als No-Free-Lunch-Theorem bezeichnet: Es gibt kein allgemein „bestes“ Verfahren!, siehe [62]

Beispiele:	Nichtlineare Funktionen	Aufwändige Funktionswertberechnung	Multikriteriell	Beschränkter Parameterraum	unlösbare Varianten	wenige Informationen (Gestalt)	viele Extrema
Modellabgleich MKS-Modell	X	X	(X)		(X)	X	
Modellabgleich CFD-Modell	X	X	(X)		(X)	X	(X)
Modellabgleich Hydraulik-Modell	X	X	(X)			X	
Kettentriebs-Layout-Optimierung	X		X	(X)	X	X	(X)
Auslegung unrunder Kettenräder	X	X	X		(X)	X	(X)
Dimensionierung Kurbelwelle	X	(X)	X	X		X	(X)
Optimierung Ventiltrieb	X	(X)	X	(X)		X	(X)
Optimierung Kanalgeometrie	X	X	X			X	X
Optimierung Kurbeltriebsgeometrie	X		X	X		X	X
Abstimmung Materialmodell für FEM	X	X				X	
Getriebeabstimmung	X		(X)	X	X	X	X

Tabelle 1.1.: Beispiele komplexer technischer Optimierungsaufgaben aus dem Bereich der Motorenentwicklung

**Möglichst viel „Information“ aus berechneten Punkten gewinnen:** Da die Anzahl der Zielfunktionswertberechnungen stark eingeschränkt sein kann, haben Verfahren, die Informationen wie Parameterabhängigkeiten oder andere Zusammenhänge erkennen und verarbeiten können, in der Regel Vorteile gegenüber Verfahren, die nur den Funktionswert nutzen.

**Keine zusätzlichen Informationen benötigen:** Gradienten und andere Informationen, die beispielsweise bei analytisch vorliegenden Zielfunktionen leicht ermittelt werden können, stehen bei Blackbox-Problemen<sup>4</sup> nicht zur Verfügung und könnten nur numerisch und mit großem Aufwand bestimmt werden.

**Mehrere Ziele berücksichtigen:** Bei vielen Problemstellungen ist es wichtig, mehrere konfliktäre Ziele zu berücksichtigen.

<sup>4</sup>Über die Gestalt der Funktion ist wenig oder nichts bekannt. Der funktionale Zusammenhang wird z.B. durch eine komplexe Simulation beschrieben. siehe [28]

## 1. Einleitung

**Parallelisierbar sein:** Die Gesamtdauer einer Optimierung kann drastisch verringert werden, wenn die Funktionsauswertungen parallel ausgeführt werden. Da die heutige Computertechnik meist mehrere CPUs besitzt und Rechner leicht zu Netzwerken verknüpft werden können, ist die Parallelisierung ein wichtiger Baustein, um umfangreiche Optimierungen in akzeptablen Zeiten durchzuführen.

Abhängig von den Eigenschaften einer Optimierungsaufgabe ist es wichtig, ein geeignetes Verfahren auszuwählen und zu konfigurieren. Diese Auswahl ist nicht eindeutig, da meist mehrere unterschiedliche Verfahren gleich gut geeignet sein können.

Um positive Eigenschaften verschiedener Verfahren zu kombinieren, können hybride Optimierungsstrategien<sup>5</sup> implementiert werden. So können zum Beispiel Verfahren, die gut geeignet sind, den gesamten Zielraum zu untersuchen, mit Verfahren gekoppelt werden, die eine gute lokale Konvergenz erreichen (siehe Kapitel 4).

Moderne Computersysteme besitzen oft mehrere Rechenkerne und ermöglichen damit die parallele Verarbeitung von Berechnungen. Durch geeignete Software können auch mehrere Computer miteinander zu einem Rechencluster verbunden werden, wodurch es möglich wird, sehr viele Berechnungen parallel durchzuführen. Eine weitere Herausforderung beim Einsatz von Optimierungsmethoden besteht darin, die Berechnung der Funktionswerte geeignet zu verteilen, um die vorhandenen Ressourcen ausnutzen zu können. Besonders bei aufwändigen Funktionswertberechnungen ist dadurch eine signifikante Verkürzung der Optimierungsdauer möglich, auch wenn die eigentliche Simulation nicht parallelisiert ist.

### 1.2. Gliederung

Im Kapitel 2 werden die mathematischen Grundlagen der ein- und multikriteriellen Optimierung eingeführt. Dabei wird der Bezug zu komplexen technischen Aufgabenstellungen der Optimierung durch die Analyse der besonderen Herausforderungen, die bei der Lösung dieser Aufgaben auftreten, hergestellt. In Kapitel 3 wird eine Auswahl von Optimierungsverfahren vorgestellt, die für die Lösung von Problemstellungen dieser Aufgabenklassen sowie für den Einsatz innerhalb hybrider Optimierungsstrategien geeignet sind. Wichtige Eigenschaften dieser Verfahren sind in Tabelle 3.1 auf Seite 44 gegenübergestellt. Kapitel 4 beschäftigt sich mit hybriden Optimierungsstrategien, wobei die Strategien „*Globale Suche - Lokale Verfeinerung*“ (Abschnitt 4.2) und „*Ersatzmodellsuche*“ (Abschnitt 4.3) näher betrachtet werden. Die Eigenschaften dieser Methoden sind in Tabelle 4.1 auf Seite 61 aufgelistet. In Abschnitt 4.5 wird auf Basis dieser Strategien das hybride Optimierungsverfahren AHMOS konstruiert. Abschließend wird im Kapitel 5 das Verfahren AHMOS mit anderen Verfahren auf der Basis von Testfunktionen, mit denen Eigenschaften komplexer technischer Aufgabenstellungen simuliert werden, verglichen und bewertet sowie ein Ausblick auf mögliche Erweiterungen gegeben.

---

<sup>5</sup>manchmal auch als Meta-Heuristiken bezeichnet

### **1.3. Zielstellung**

Das Ziel dieser Arbeit besteht darin, bekannte und neue hybride Optimierungsstrategien hinsichtlich ihrer Potentiale beim Einsatz für technische Anwendungen zu untersuchen. Dabei werden insbesondere die Anforderungen von Optimierungsproblemen mit aufwändiger Funktionswertberechnung berücksichtigt. Weiterhin wird die Anwendbarkeit der Methoden auf ein- und multikriterielle Zielfunktionen analysiert.

Ausgewählte Strategien werden dann zu dem hybriden Optimierungsverfahren AHMOS zusammengeführt. Mit geeigneten Tests soll der Nutzen der gewählten hybriden Strategien nachgewiesen werden.



# 2

## Kapitel 2.

# Nichtlineare Multikriterielle Optimierung

## 2.1. Grundlagen

Eine allgemeine *Nichtlineare Multikriterielle Optimierungsaufgabe* (NLMOA) mit  $M$  Zielfunktionen,  $J$  Nebenbedingungen und  $N$  Optimierungsvariablen, die im Variablenvektor  $\mathbf{x} = (x_1, x_2, \dots, x_N)^T \in \mathbb{R}^N$  zusammengefasst werden, kann allgemein wie folgt geschrieben werden:

$$(NLMOA) \begin{cases} f_m(\mathbf{x}) \rightarrow \min & m = 1, 2, \dots, M \\ g_j(\mathbf{x}) \geq 0 & j = 1, 2, \dots, J \\ x_i^{(U)} \leq x_i \leq x_i^{(O)} & i = 1, 2, \dots, N \\ \mathbf{x} \in \mathbb{R}^N \end{cases} \quad (2.1)$$

mit  $M, J, N \in \mathbb{N}$ , wobei sowohl die Zielfunktionen<sup>1</sup> als auch die Nebenbedingungen durch nichtlineare Funktionen gegeben sind. Die Optimierungsvariablen  $x_i$  sind jeweils durch eine untere Grenze  $x_i^{(U)}$  und eine obere Grenze  $x_i^{(O)}$  beschränkt. Ohne Einschränkung der Allgemeinheit werden im Folgenden nur Minimierungsaufgaben mit Ungleichheitsnebenbedingungen betrachtet, da die Maximierung von  $f(\mathbf{x})$  äquivalent zur Minimierung von  $-f(\mathbf{x})$  ist und jede Gleichung  $h(\mathbf{x}) = 0$  durch zwei Ungleichungen  $h(\mathbf{x}) \geq 0 \wedge h(\mathbf{x}) \leq 0$  dargestellt werden kann.

**Definition:** Gegeben sei eine NLMOA<sup>2</sup>.

Die Menge aller  $\mathbf{x} \in \mathbb{R}^N$ , die in den gegebenen Intervallgrenzen liegen und die Nebenbedingungen erfüllen, werden als *zulässige Menge*

<sup>1</sup>Einkriterielle Optimierungsaufgaben können als Spezialfall von (2.1) mit  $M = 1$  betrachtet werden.

<sup>2</sup>in der Literatur auch bezeichnet als Vektoroptimierungsproblem

## 2. Nichtlineare Multikriterielle Optimierung

$$S = \left\{ \mathbf{x} \in \mathbb{R}^N \mid \begin{array}{ll} g_j(\mathbf{x}) \geq 0 & j = 1, 2, \dots, J \\ x_i^{(U)} \leq x_i \leq x_i^{(O)} & i = 1, 2, \dots, N \end{array} \right\} \quad (2.2)$$

bezeichnet.

Die Menge der Lösungen

$$F_S = \left\{ (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))^T \in \mathbb{R}^M \mid \forall \mathbf{x} \in S \right\} \quad (2.3)$$

wird als Ergebnismenge bezeichnet.

Während eine einkriterielle Optimierungsaufgabe eine oder mehrere Lösungen mit dem optimalen Funktionswert  $f(\mathbf{x}^*) \in F_S \subseteq \mathbb{R}$  besitzt, der ein Minimum (Minimierungsaufgabe) oder ein Maximum (Maximierungsaufgabe) der Zielfunktion ist, besteht die Lösung einer multikriteriellen Optimierungsaufgabe aus einer Menge an möglichen Kompromissen - der PARETO-Menge.

**Definition:** Gegeben sei eine NLMOA mit der zulässigen Menge  $S$ .

- Ein Punkt  $\mathbf{x} \in S$  *dominiert*  $\mathbf{y} \in S$ , kurz:  $\mathbf{x} \prec \mathbf{y}$ , falls gilt:  
 $\forall m \in \{1, 2, \dots, M\} : f_m(\mathbf{x}) \leq f_m(\mathbf{y})$  und  
 $\exists \hat{m} \in \{1, 2, \dots, M\} : f_{\hat{m}}(\mathbf{x}) < f_{\hat{m}}(\mathbf{y})$ .
- Ein Punkt  $\mathbf{x} \in S$  heißt *nichtdominiert* bezüglich einer Menge  $T \subseteq S$  genau dann, wenn kein Element in  $T$  existiert, das  $\mathbf{x}$  dominiert:  $\nexists \hat{\mathbf{x}} \in T : \hat{\mathbf{x}} \prec \mathbf{x}$ .
- Ein Punkt ist *PARETO-optimal* genau dann, wenn  $\mathbf{x}$  nichtdominiert bezüglich  $S$  ist.
- Eine Menge  $X \subseteq S$  heißt *PARETO-Menge* genau dann, wenn sie nur aus nichtdominierten Punkten besteht.
- Die Menge  $P = \left\{ (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))^T \in \mathbb{R}^M \mid \forall \mathbf{x} \in X \right\}$  wird als *PARETO-Front* bezeichnet.

Das Ziel eines Verfahrens zur Lösung einer multikriteriellen Optimierungsaufgabe ist somit die Bestimmung der PARETO-Menge, aus der dann eine beste Lösung<sup>3</sup>, d.h. ein geeigneter Kompromiss, ausgewählt wird. Ein geeigneter Algorithmus sollte also einerseits schnell zur PARETO-Menge konvergieren und andererseits die Menge möglichst gleichmäßig approximieren.

<sup>3</sup>eine beste Lösung wird von der bestimmten Anwendung bestimmt und ist z.B. abhängig von der Gewichtung der Ziele und anderen Bewertungskriterien



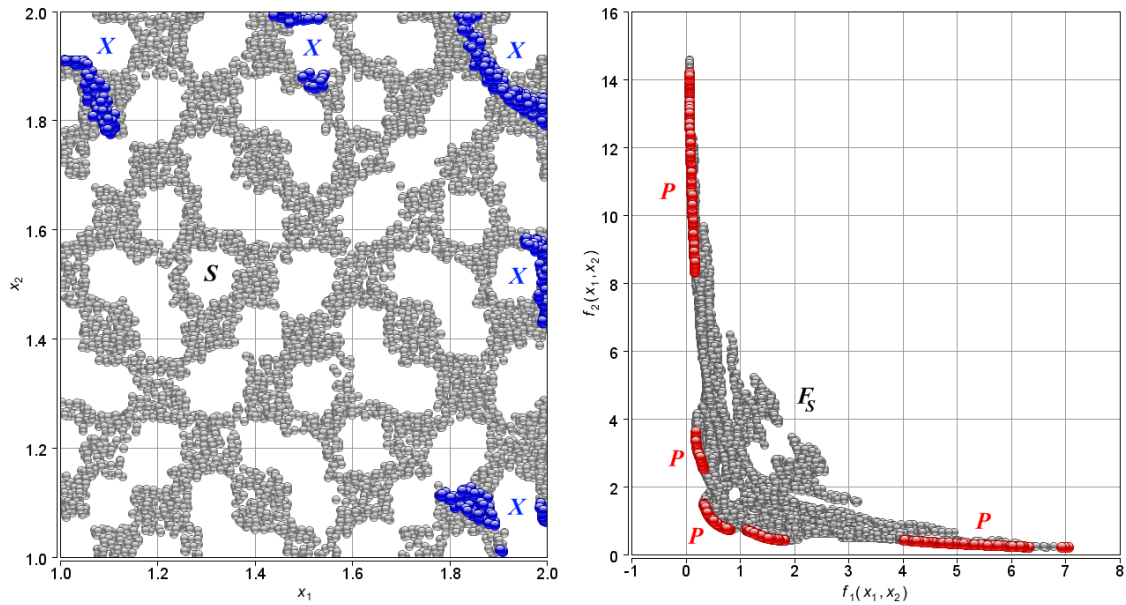


Abbildung 2.1.: Beispiel mit 2 Variablen und 2 Zielen (Minimierung): zulässige Menge  $S$ , Ergebnismenge  $F_S$ , PARETO-Menge  $X$ , PARETO-Front  $P$

**Beispiel:** Abbildung<sup>4</sup> 2.1 zeigt die zulässige Menge  $S$ , die Ergebnismenge  $F_S$ , die PARETO-Menge<sup>5</sup>  $X$  und die PARETO-Front<sup>6</sup>  $P$  der Testaufgabe MINER-M-1<sup>7</sup> mit zwei Variablen, zwei zu minimierenden Zielen und zwei Nebenbedingungen:

$$f_{\text{Miner-M-1}} \begin{cases} 2 \cdot \left( \frac{x_1}{x_2^2} \right) + \frac{\cos(\pi x_1(x_1+x_2))}{10} \rightarrow \min \\ 4 \cdot \left( \frac{x_2}{x_1^2} \right) + \frac{\cos(\pi x_2(x_1+x_2))}{10} \rightarrow \min \\ \sin(30x_1) \sin(30x_2) < \frac{1}{8} \\ \sin(20x_1) \sin(20x_2) < \frac{1}{2} \\ 1 \leq x_i \leq 2 \quad i = 1, 2 \\ \mathbf{x} \in \mathbb{R}^2. \end{cases} \quad (2.4)$$

<sup>4</sup>Bei den dargestellten Mengen handelt es sich jeweils um eine Approximation der Menge, die mit Hilfe eines Optimierungsverfahrens ermittelt wurde.

<sup>5</sup>blau hervorgehoben

<sup>6</sup>rot hervorgehoben

<sup>7</sup>siehe Abschnitt 5.2.3

## 2.2. Herausforderungen bei der Optimierung von technischen Problemstellungen

Im Entwicklungsprozess technischer Bauteile und Funktionssysteme spielt der Einsatz effektiver mathematischer Optimierungsmethoden eine wichtige Rolle. Aufgaben mit Eigenschaften, wie sie in Abschnitt 1 aufgezählt wurden, erfordern gut angepasste Optimierungsverfahren, um eine Lösung mit vertretbarem Aufwand finden zu können. Dafür ist es wichtig, einige diese Eigenschaften genauer zu analysieren, um geeignete Methoden auswählen und entwickeln zu können.

### 2.2.1. Aufwändige, restringierte Blackbox-Probleme in der Technik

Optimierungsprobleme, bei denen über die Zielfunktionen nur wenige oder keine Informationen vorliegen, werden als *Blackbox-Probleme*<sup>8</sup> bezeichnet. Wie aus der Bezeichnung ableitbar, handelt es sich hierbei um Probleme, bei denen die Variablen als Parameter in ein parametrisiertes Modell<sup>9</sup> eingehen, das dann berechnet wird und dessen Ergebnisse zurückgegeben werden. Insbesondere dann, wenn hinter dieser Funktionsauswertung komplexe numerische Simulationen stehen, wie dies bei technischen Anwendungen meist der Fall ist, spricht man von *aufwändigen technischen Blackbox-Problemen*. Die zulässige Menge  $S$  kann zusätzlich von einfachen oder komplexen Nebenbedingungen eingeschränkt sein.

**Definition:** *Technische Blackbox-Probleme* sind durch folgende Eigenschaften charakterisiert (siehe [28] und [58]):

- die Gesamtzahl an Funktionswertbestimmungen ist aus finanziellen, zeitlichen oder Kapazitätsgründen eingeschränkt oder stark eingeschränkt,
- nur eine begrenzte Anzahl an Zielfunktionsauswertungen kann zeitgleich vorgenommen werden, das heißt, eine Parallelisierung ist nur eingeschränkt möglich,
- wiederholte Berechnungen ergeben gleiche oder sehr ähnliche Ergebnisse (geringes Rauschen),
- in der Regel sind große Verbesserungen gegenüber dem Ausgangsstand erzielbar,
- die Suchlandschaft ist lokal relativ glatt<sup>10</sup>, kann aber eine große Zahl von lokalen Optima besitzen,
- die Dimension des Suchraumes ist klein bis mittel (Anzahl der Variablen liegt in der Regel zwischen 5 und 30)

---

<sup>8</sup>siehe auch [28, 58]

<sup>9</sup>z.B. ein mathematisch-physikalisches Modell, das nur numerisch gelöst werden kann

<sup>10</sup>hinreichend oft differenzierbar

## 2.2. Herausforderungen bei der Optimierung von technischen Problemstellungen

- die Probleme enthalten mehrere, sich meist widersprechende Ziele.
- es treten vielfach nichtlineare Nebenbedingungen auf, die den Definitionsbereich beschränken,
- die zulässige Menge  $S$  ist oft nicht einfach zusammenhängend,
- im Inneren der zulässigen Menge  $S$  gibt es einfach zusammenhängende beschränkte Punktmengen, die nicht zum Definitionsbereich gehören (technisch nicht realisierbare Varianten),
- einzelne Variablen können nur diskrete Zustände annehmen,
- es stehen keine Gradienteninformationen der Zielfunktionen zur Verfügung,
- eine analytische Beschreibung der Ziele ist nicht möglich.

Nicht jedes aufwändige Blackbox-Problem wird alle diese Eigenschaften aufweisen, in der Praxis treten meistens unterschiedliche Kombinationen davon auf. Im Folgenden wird eine Auswahl wichtiger Eigenschaften näher betrachtet, die in dieser Arbeit besonders von Bedeutung sind.

### 2.2.1.1. Mehrere widersprüchliche Ziele

Bei einem großen Teil von Optimierungsaufgaben, die zur Lösung von Problemstellungen aus der Technik genutzt werden, ist es nicht möglich oder nicht sinnvoll, nur ein einzelnes Ziel zu betrachten, sondern es ist vielmehr erforderlich, mehrere, in der Regel konfliktäre Ziele, zu berücksichtigen. Dies soll an den folgenden Beispielen verdeutlicht werden:

**Kurbelwellenauslegung:** Um eine Kurbelwelle zu optimieren, ist es nicht möglich, nur ein Ziel zu betrachten. Wird ihre Masse minimiert, indem zum Beispiel die Gegengewichte verkleinert werden, kann das dazu führen, dass die Unwucht der Welle größer wird und damit auch die Kräfte, die auf die Lager wirken. Werden hingegen die Lagerkräfte minimiert, so müssen dazu meist die Ausgleichsgewichte vergrößert werden, womit aber auch eine Erhöhung der Masse verbunden ist. Eine sinnvolle Optimierung muss also in irgendeiner Weise beide Ziele berücksichtigen und wird als Endergebnis einen Kompromiss zwischen beiden Zielen anstreben.

**Modellabgleich eines Dynamikmodells:** Eine typische Aufgabe für den Einsatz von Optimierungsmethoden ist der automatisierte Modellabgleich, bei dem die Parameter eines Simulationsmodells so abgestimmt werden sollen, dass es Eigenschaften, die beispielsweise mit einer Messung vorgegeben sind, erfüllt. Bei der Abstimmung von Dynamikmodellen ist es dabei sinnvoll, den Abgleich für einzelne FOURIER-Ordnungen getrennt durchzuführen. Das führt wieder zu einer multikriteriellen Optimierungsaufgabe, da eine Übereinstimmung der einzelnen

## 2. Nichtlineare Multikriterielle Optimierung

Ordnungen in der Regel nicht gleichzeitig erreicht wird und somit ein Kompromiss gefunden werden muss.

Optimierungsmethoden, die zur Lösung von Aufgaben mit mehreren Zielen geeignet sind, müssen die PARETO-Menge (oder zumindest einen speziellen Teil davon) näherungsweise bestimmen können. Dabei ist es wichtig, dass die Realisierungen der gefundenen Kompromisse möglichst dicht und gleichmäßig verteilt an der PARETO-Front liegen.

### 2.2.1.2. Unlösbare Varianten

Für *unlösbare Varianten* ist es nicht möglich, die Zielfunktionen zu berechnen. Im Gegensatz zu unzulässigen Varianten, für die zwar die Zielfunktionen berechnet werden können, welche aber die Restriktionen verletzen, sind sie für die Optimierung unbrauchbar. Verschiedene Ursachen können dafür verantwortlich sein, dass unlösbare Varianten entstehen:

**Bestimmte Parameterkonstellationen führen zu technisch nicht realisierbaren Varianten:** Beispielsweise kann die Änderung von geometrischen Parametern dazu führen, dass zwei Bauteile miteinander kollidieren, obwohl jedes einzelne Teil für sich betrachtet zulässig ist. Die Baugruppe als Ganzes ist dann nicht umsetzbar.

**Die Simulationssoftware kann ein bestimmtes Modell nicht berechnen:** Das kann darin begründet sein, dass die Software nicht immer stabil arbeitet. So ist es zum Beispiel möglich, dass die Anfangsbedingungen einer Dynamiksimulation nicht bestimmt werden können und somit die Simulation gar nicht gestartet wird. Ein weiterer Grund kann sein, dass die Simulation aufgrund numerischer Probleme vorzeitig abbricht und dadurch die Ergebnisse nicht zur Verfügung stehen.

Das Auftreten unlösbarer Varianten ist in der Praxis nicht oder bestenfalls mit sehr großem Aufwand vorhersehbar, so dass es nur selten möglich ist, durch eine Umformulierung der Optimierungsaufgabe zu verhindern, dass solche Fälle auftreten. Geeignete Optimierungsverfahren dürfen sich durch das Auftreten unlösbarer Varianten nicht signifikant behindern lassen. So muss zum Beispiel verhindert werden, dass vom Algorithmus immer wieder die gleiche oder sehr ähnliche Varianten ausgewertet werden sollen, wenn diese als unlösbar erkannt werden. Viele stochastische Verfahren, wie zum Beispiel Evolutionäre Verfahren, können von ihrem Ansatz her gut damit umgehen, für andere Verfahren muss eine besondere Behandlung solcher Varianten erfolgen.

### 2.2.1.3. Viele lokale Extrema

Nichtlineare Optimierungsprobleme, bei denen die Zielfunktionen durch komplexe numerische Simulationen gegeben sind, sind meist global nicht glatt<sup>11</sup> und monoton, sondern haben viele lokale Extrema<sup>12</sup>. Werden lokal arbeitende Optimierungsverfahren<sup>13</sup> verwendet, konvergieren diese, abhängig vom Startpunkt, im nächstgelegenen lokalen Extrempunkt. Ziel der Optimierung sollte es aber sein, das globale Extremum oder zumindest eine gute Näherung davon zu finden.

Geeignete Verfahren sollten so arbeiten, dass sie möglichst schnell konvergieren, ohne in einem lokalen Extremum hängen zu bleiben.

Eine weitere Herausforderung bei MOA<sup>14</sup> kann sein, dass es mehrere Lösungsgebiete gibt, die mit unterschiedlichen Parameterwerten ähnlich gute Ergebnisse liefern, so dass die PARETO-Menge möglicherweise aus disjunkten Teilmengen besteht. Bei der Lösung einer solchen Aufgabe ist es wichtig, dass so ein Gebiet nicht vorzeitig verworfen wird, sondern alle Teilgebiete untersucht werden.

### 2.2.1.4. Schmale und nichtzusammenhängende Restriktionsbereiche

Bei technischen Aufgabenstellungen ist es möglich, dass mehrere, zum Teil auch sehr komplexe, Nebenbedingungen eingehalten werden müssen, um sinnvolle Ergebnisse zu erhalten. Besonders dann, wenn Ergebnisse der Simulation in die Restriktionen einfließen, kann es vorkommen, dass die zulässige Menge  $S$  sehr „schmal“ und oft auch nicht zusammenhängend ist. Dadurch kann es für das Optimierungsverfahren schwierig sein, überhaupt zulässige Lösungen zu finden.

Geeignete Verfahren müssen in der Lage sein, auch mit unzulässigen Lösungen umzugehen und dabei aber im Laufe der Optimierung dafür sorgen, dass die Lösungen sich schnellstmöglich in Richtung des zulässigen Bereiches bewegen.

### Das „Alpen-Problem<sup>15</sup>“

Besteht die zulässige Menge  $S$  aus mehreren disjunkten Teilmengen oder existieren disjunkte Bereiche mit guten Lösungen in der Lösungsmenge, ist es möglich, dass die Optimierung sich in einem Teilgebiet festsetzt und andere Teilgebiete gar nicht untersucht werden. Bei einem neuen Start der Optimierung kann es dann passieren, dass ein anderes Gebiet als Erstes erreicht wird und die Optimierung in diesem Gebiet verharrt. Die Ergebnisse der Optimierung können insbesondere dann sehr unterschiedlich sein, wenn stochastische Verfahren eingesetzt werden, wodurch die Suche zufällig das eine oder das andere Gebiet findet und danach dort verharrt.

<sup>11</sup>hinreichend oft differenzierbar

<sup>12</sup>das gilt analog auch für die PARETO-Menge bei MOA

<sup>13</sup>zum Beispiel Gradientenverfahren

<sup>14</sup>Multikriterielle Optimierungsaufgabe

<sup>15</sup>„Beginnt man auf einem Gipfel den Abstieg ins Tal, wird ein abgeschiedenes Dorf erreicht. Startet man den Abstieg in einer anderen Richtung, wird ein weiteres, ebenfalls abgeschiedenes Dorf gefunden. Die Dörfer können untereinander nur erreicht werden, wenn man erneut einen Weg über den Gipfel wählt.“

## 2. Nichtlineare Multikriterielle Optimierung

Verfahren, die für Problemstellungen dieser Art geeignet sind, benötigen Strategien, die zum einen mehrere disjunkte Bereiche betrachten können und im Laufe der Optimierung auch nach neuen, bisher nicht bekannten Gebieten suchen.

### 2.2.2. Parallelisierung

Moderne Computersysteme, die mehrere CPU bzw. Rechenkernen besitzen oder aus mehreren verbundenen Rechnern bestehen, bieten die Möglichkeit, viele Berechnungen parallel auszuführen. Besonders bei aufwändigen Optimierungsproblemen spielt die Parallelisierung eine signifikante Rolle, da dadurch die Probleme überhaupt erst lösbar und die Rechenzeiten drastisch verkürzt werden können.

Viele Optimierungsverfahren, wie zum Beispiel die Klasse der *naturinspirierten Verfahren*<sup>16</sup>, die als Grundlage stets eine Menge<sup>17</sup> von Varianten betrachten, sind leicht für eine mittlere Anzahl von Prozessoren<sup>18</sup> parallelisierbar. Es existieren auch Anpassungen für sehr viele CPU, diese Methoden werden in dieser Arbeit jedoch nicht näher betrachtet. Verfahren, die in einem Optimierungsschritt stets auch nur einen neuen Punkt ermitteln, beispielsweise *adaptive Ersatzmodellverfahren* und die meisten *deterministischen Verfahren*, benötigen angepasste Erweiterungen, um mehrere Schritte parallel berechnen zu können. Oft sinkt dadurch aber die Effektivität dieser Verfahren, so dass ein sinnvoller Kompromiss zwischen Parallelisierung und Effektivität gefunden werden muss.

In der praktischen Anwendung sind der Parallelisierung Grenzen gesetzt. So können die verfügbaren Rechenkerne, aber auch die Verfügbarkeit von Lizenzen für die zur Funktionsauswertung benötigten Simulationsprogramme den Grad der Parallelisierung einschränken.

---

<sup>16</sup>siehe [9, 47]

<sup>17</sup>auch als Population bezeichnet

<sup>18</sup>weniger als 100 CPUs

# 3

## Kapitel 3.

---

# Ausgewählte Optimierungsverfahren

Als Ergebnis zahlreicher Forschungen sind eine Vielzahl von Methoden entwickelt worden, um nichtlineare Optimierungsaufgaben zu lösen<sup>1</sup>. Während deterministische Verfahren auf einem reproduzierbaren Algorithmus aufgebaut sind, nutzen stochastische Verfahren den Zufall<sup>2</sup> und heuristische Ansätze<sup>3</sup>, um Optima zu finden. Dabei ist es aber nicht möglich, *ein bestes Verfahren* für die Lösung aller Optimierungsprobleme<sup>4</sup> anzugeben, weshalb die Auswahl des Optimierungsalgorithmus stets von der zu lösenden Aufgabe abhängig ist.

Mit modernen hybriden Ansätzen (siehe Kapitel 4) wird die Zielstellung verfolgt, durch die Kombination verschiedener Methoden und Strategien und den damit verbundenen Vorteilen, neue Verfahren zu konstruieren, die spezielle Problemstellungen besser lösen können oder vielseitig für eine größere Menge von Problemen einsetzbar sind.

## 3.1. Überblick

In Abbildung 3.1 sind Verfahren der nichtlinearen Optimierung dargestellt. Auch wenn diese Darstellung nicht vollständig ist, werden die wesentlichen Klassen und wichtige Vertreter dieser Klassen vorgestellt. Hybride Strategien sind im eigentlichen Sinn keine eigene Verfahrensklasse, da sie eine Kombination unterschiedlicher Verfahren darstellen, können aber als solche betrachtet werden, da sie neue Eigenschaften aufweisen, die sie von deterministischen und stochastischen Verfahren abgrenzen.

Die naturinspirierten Verfahren als Vertreter der stochastischen Verfahren werden etwas näher betrachtet, da sie im Rahmen dieser Arbeit als Basis der hybriden Verfahren eingesetzt werden.

---

<sup>1</sup>siehe z.B. [4, 12, 58, 49]

<sup>2</sup>Bei Bedarf kann Reproduzierbarkeit durch Verwendung von Pseudozufallszahlen erreicht werden.

<sup>3</sup>zum Beispiel Vorgänge aus der Biologie oder Physik → naturinspirierte Verfahren

<sup>4</sup>auch als No-Free-Lunch-Theorem bezeichnet, siehe [62]

### 3. Ausgewählte Optimierungsverfahren

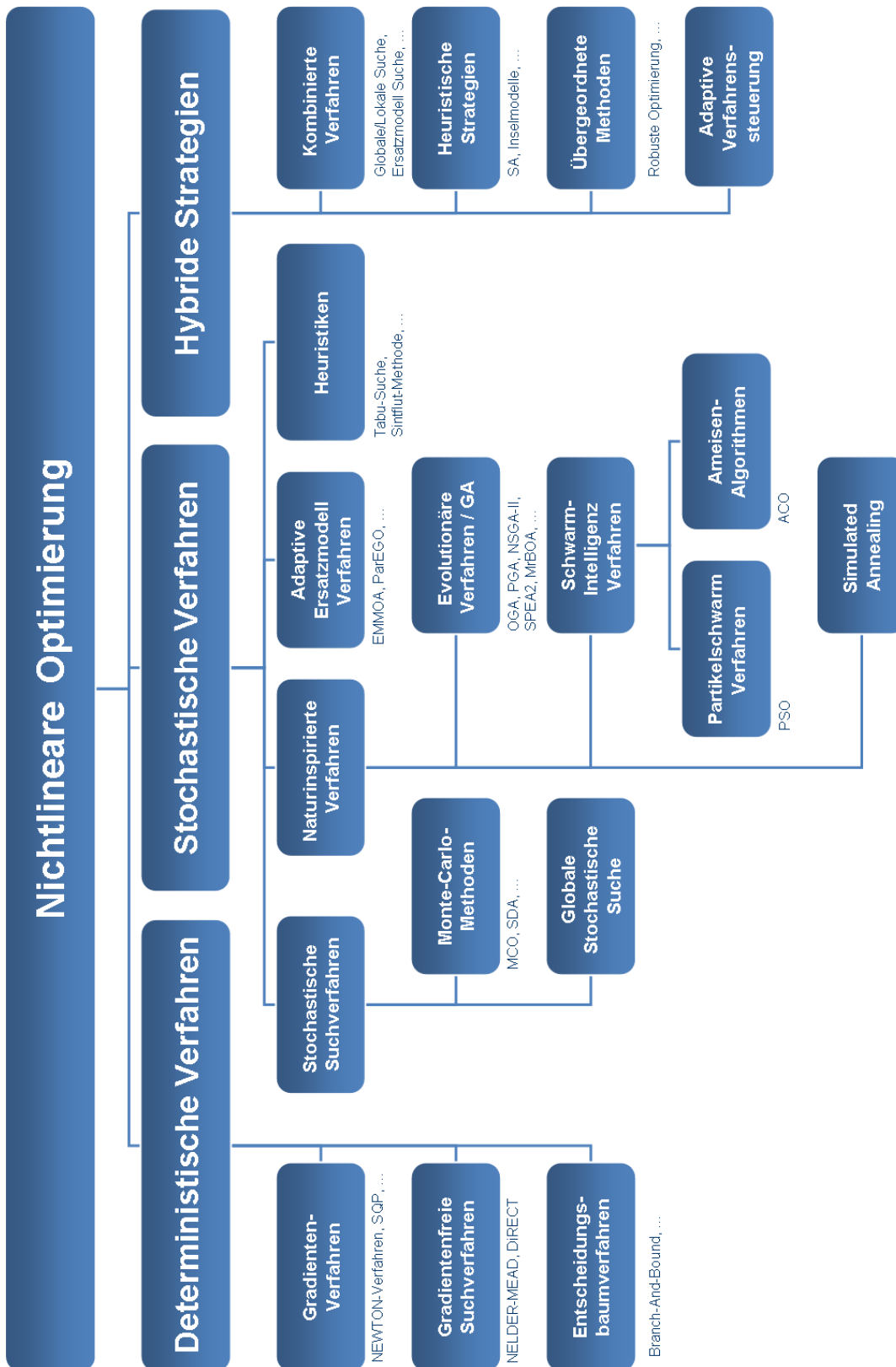


Abbildung 3.1.: Übersicht Nichtlineare Optimierungsverfahren



## 3.2. Deterministische Verfahren und deren multikriterielle Erweiterungen

*Deterministische Verfahren*, manchmal auch als *Exakte Verfahren* bezeichnet, sind Algorithmen, die sich durch einen klaren algorithmischen Ablauf ohne stochastische Einflüsse oder Heuristiken auszeichnen. Sie sind damit bei gleichen Ausgangsbedingungen stets eindeutig reproduzierbar. Unter geeigneten Voraussetzungen arbeiten diese Verfahren sehr effektiv. Da es sich bei diesen Methoden um Lösungsverfahren für einkriterielle Optimierungsprobleme handelt, können NLMOA nur durch die Nutzung der klassischen Methoden, die die multikriterielle Optimierungsaufgabe in eine einkriterielle Aufgabe überführen, gelöst werden (siehe Abschnitt 3.2.3)<sup>5</sup>.

### 3.2.1. Gradientenverfahren

Gradientenverfahren<sup>6</sup> werden häufig zur Lösung nichtlinearer, einkriterieller Optimierungsaufgaben eingesetzt, deren Zielfunktion mindestens einmal stetig differenzierbar ist. Der Gradient

$$\nabla f(\mathbf{x}) = \begin{pmatrix} \frac{\partial}{\partial x_1} f(\mathbf{x}) \\ \vdots \\ \frac{\partial}{\partial x_N} f(\mathbf{x}) \end{pmatrix}$$

einer Funktion  $f(\mathbf{x})$  mit  $\mathbf{x} = (x_1, \dots, x_N)$  gibt dabei die Richtung des steilsten Anstiegs von  $f$  an der Stelle  $\mathbf{x}$  an. Ausgehend von einem zulässigen Startpunkt schreitet man in Richtung des negativen Gradienten (d.h. in die Richtung des steilsten Abstiegs) voran<sup>7</sup>, bis keine weitere Verbesserung mehr erzielt werden kann oder der Rand des Lösungsraumes erreicht ist. Von dieser Stelle aus wird dann der nächste Iterationsschritt gestartet bis die Konvergenzkriterien erfüllt sind. Für OA mit Nebenbedingungen muss sichergestellt werden, dass der gewählte Schritt nicht aus dem zulässigen Bereich herausführt. Liegt ein konvexes Optimierungsproblem vor, wird die optimale Lösung durch dieses Vorgehen mit beliebiger Genauigkeit erreicht<sup>8</sup>, andernfalls liefern Gradientenverfahren in der Regel nur lokale Optima.

Der prinzipielle Ablauf eines Gradientenverfahrens (ohne Behandlung von Nebenbedingungen) ist in Algorithmus 3.1 dargestellt<sup>9</sup>.

Abhängig von der Gestalt der zu lösenden Optimierungsaufgabe, können verschiedene spezielle Verfahren, wie das NEWTON-Verfahren, das CG-Verfahren<sup>10</sup>, der SQP<sup>11</sup>-Algorithmus

<sup>5</sup>siehe [4]

<sup>6</sup>auch bezeichnet als Verfahren des steilsten Abstiegs

<sup>7</sup>Moderne Gradientenverfahren wählen als Abstiegsrichtung oft nicht die direkte Richtung des negativen Gradienten (siehe [4]).

<sup>8</sup>siehe auch [18]

<sup>9</sup>siehe [42]

<sup>10</sup>Verfahren der konjugierten Gradienten, [4]

<sup>11</sup>Sequential Quadratic Programming; [4]

### 3. Ausgewählte Optimierungsverfahren

---

**Algorithmus 3.1** Gradientenverfahren

---

**Eingabe** : OA

Wähle Startvektor  $\mathbf{x}^{(0)}$

$k = 0$

**while**  $\|\nabla f(\mathbf{x}^{(k)})\| > \varepsilon$  **do**

    Wähle Abstiegsrichtung:  $\mathbf{d}^{(k)} = -\nabla f(\mathbf{x}^{(k)})$

    Bestimme die Schrittweite  $t^{(k)}$

$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)}\mathbf{d}^{(k)}$

$k = k + 1$

**end**

---

usw. effektiv eingesetzt werden<sup>12</sup>. Voraussetzung ist dabei aber, dass die ersten und zum Teil auch die zweiten Ableitungen existieren und effektiv bestimmt werden können. Dies stellt beim Einsatz dieser Optimierungsverfahren für komplexe technische Probleme (siehe Abschnitt 2.2) oft eine Schwierigkeit dar, da die Gradienteninformationen nicht zur Verfügung stehen und nur aufwändig numerisch ermittelt werden können. Insbesondere dann, wenn für die Simulation zur Funktionsauswertung recht oft Varianten unlösbar sind (siehe Abschnitt 2.2), ist eine numerische Bestimmung der Gradienten kaum möglich.

#### 3.2.2. Gradientenfreie Verfahren

Bei vielen Optimierungsproblemen ist die Zielfunktion nicht überall differenzierbar oder die Berechnung der Ableitungen erfordert einen großen numerischen Aufwand<sup>13</sup>. In diesen Fällen bieten sich Optimierungsverfahren an, die ohne die Gradienteninformationen auskommen. Im Folgenden werden ein lokales und ein globales Verfahren dargestellt, die sich gut zur Lösung nichtlinearer EOA eignen.

##### 3.2.2.1. Nelder-Mead-Simplex-Verfahren<sup>14</sup>

Das NELDER-MEAD-Simplex-Verfahren, auch bekannt unter dem Namen *Downhill-Simplex-Verfahren*, benutzt eine Grundstruktur, den Simplex, zur Lösung von Optimierungsaufgaben.

---

<sup>12</sup>siehe auch [53, 4]

<sup>13</sup>z.B. wenn für die Funktionswertberechnung eine komplexe Simulation ausgeführt werden muss

<sup>14</sup>siehe auch [4, 55]

### 3.2. Deterministische Verfahren und deren multikriterielle Erweiterungen

**Definition:** Seien  $\mathbf{x}^{(0)}, \dots, \mathbf{x}^{(n)} \in \mathbb{R}^N$  linear unabhängige Vektoren. Dann heißt die Menge

$$\mathbb{S} = \left\{ \sum_{i=0}^n \lambda_i \mathbf{x}^{(i)} \mid \lambda_i \geq 0; i = 0, \dots, n; \sum_{i=0}^n \lambda_i = 1 \right\}$$

$n$ -dimensionales Simplex mit den Ecken  $\mathbf{x}^{(0)}, \dots, \mathbf{x}^{(n)}$ .

Der Punkt  $\mathbf{s}^{(j)}$  mit  $j \in \{1, \dots, n\}$  mit

$$\mathbf{s}^{(j)} = \frac{1}{n} \sum_{\substack{i=0 \\ i \neq j}}^n \mathbf{x}^{(i)}$$

heißt Schwerpunkt bezüglich  $\mathbf{x}^{(j)}$ .

Die Eckpunkte des Simplex sind Lösungen der OA. Für eine Aufgabe mit  $N$  Variablen wird ein Simplex mit  $n = N + 1$  Ecken verwendet<sup>15</sup>. Ausgehend von einer Startlösung wird in jedem Iterationsschritt des Algorithmus die schlechteste Ecke des Simplex jeweils durch einen besseren Punkt ersetzt. Zu dessen Bestimmung werden die folgenden drei Konstruktionsoperationen benutzt:

**Reflektion:** Der schlechteste Punkt  $\mathbf{x}^{(j)}$  mit  $j \in \{1, \dots, n\}$  wird am Schwerpunkt  $\mathbf{s}^{(j)}$  reflektiert, man erhält den reflektierten Punkt  $\mathbf{x}_R = \mathbf{s}^{(j)} + \gamma (\mathbf{s}^{(j)} - \mathbf{x}^{(j)})$  mit  $0 < \gamma \leq 1$ .

**Expansion:** Der reflektierte Punkt  $\mathbf{x}_R$  wird noch weiter in Richtung  $\mathbf{x}_R - \mathbf{s}^{(j)}$  verschoben und man erhält der expandierten Punkt  $\mathbf{x}_E = \mathbf{s}^{(j)} + \beta (\mathbf{x}_R - \mathbf{s}^{(j)})$  mit  $\beta > 1$ .

**Kontraktion:** Bei der Kontraktion unterscheidet man drei verschiedene Typen:

- *innere Kontraktion* in Richtung  $\mathbf{x}^{(j)} - \mathbf{s}^{(j)}$  führt zu dem Punkt  $\mathbf{x}_C = \mathbf{s}^{(j)} + \alpha (\mathbf{x}^{(j)} - \mathbf{s}^{(j)})$  mit  $0 < \alpha < 1$ ,
- *äußere Kontraktion* in Richtung  $\mathbf{x}_R - \mathbf{s}^{(j)}$  führt zu dem Punkt  $\mathbf{x}_C = \mathbf{s}^{(j)} + \alpha (\mathbf{x}_R - \mathbf{s}^{(j)})$  mit  $0 < \alpha < 1$  und
- *Schrumpfen oder totale Kontraktion* bzgl.  $\mathbf{x}^{(k)}$  mit  $k \in \{1, \dots, n\}$ , wobei alle Punkte  $\mathbf{x}^{(i)}$  mit  $i = 0, \dots, n, i \neq k$  durch die Punkte

$$\hat{\mathbf{x}}^{(i)} = \mathbf{x}^{(i)} + \frac{1}{2} (\mathbf{x}^{(k)} - \mathbf{x}^{(i)})$$

ersetzt werden. Als Kontraktionszentrum wird dabei der beste Punkt  $\mathbf{x}^{(k)}$  des Simplex verwendet.

<sup>15</sup>man kann den Simplex somit als  $(N + 1)$ -dimensionales Polyeder betrachten

### 3. Ausgewählte Optimierungsverfahren

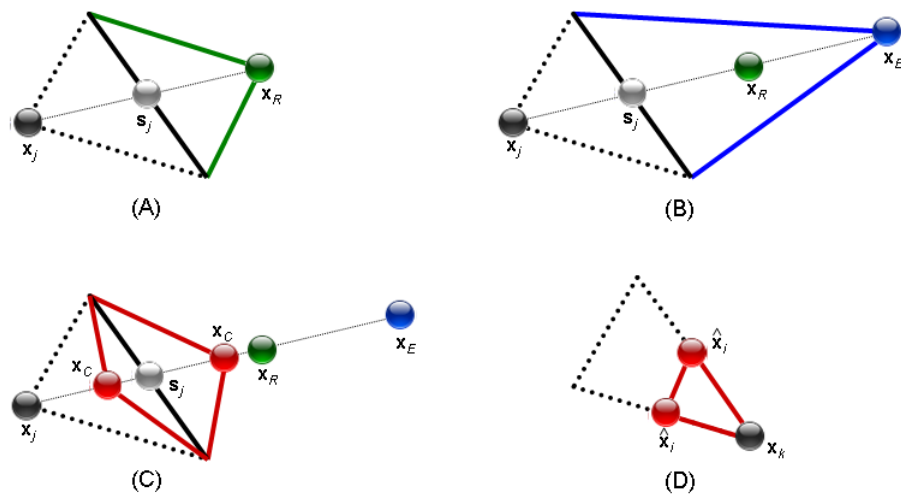


Abbildung 3.2.: NELDER-MEAD-Operationen an einem Beispiel mit 2 Variablen

(A) ... Reflektion (B) ... Expansion (C) ... Kontraktion (D) ... Schrumpfen

In Abbildung 3.2 werden die Konstruktionsoperationen an einem Beispiel mit zwei Variablen verdeutlicht. Der prinzipielle Verlauf des Verfahrens ist in Algorithmus 3.2 dargestellt.

Das Verfahren arbeitet lokal, d.h. es konvergiert, abhängig vom Startsimplex und den vorgegebenen Reflektion-, Expansions- und Kontraktionsfaktoren, zum nächsten lokalen Minimum.

---

#### Algorithmus 3.2 NELDER-MEAD-Verfahren

---

**Eingabe :** OA

Startsimplex erzeugen

**while** Abbruchbedingungen nicht erfüllt **do**

    Finde: schlechteste  $x^{(j)}$  und beste  $x^{(k)}$  Ecke des Simplex

    Berechne Schwerpunkt  $s^{(j)}$  des Simplex bezüglich  $x^{(j)}$

    Reflektiere  $x^{(j)}$  an  $s^{(j)} \rightarrow x_R$

**if**  $x_R$  ist besser als alle anderen Ecken **then**

        | Führe Expansion aus  $\rightarrow x_E$

**else if**  $x_R$  ist größer als andere Ecken **then**

        | Führe innere/äußere Kontraktion aus  $\rightarrow x_C$

**else**

        | Führe totale Kontraktion aus

**end**

**end**

---

### 3.2. Deterministische Verfahren und deren multikriterielle Erweiterungen

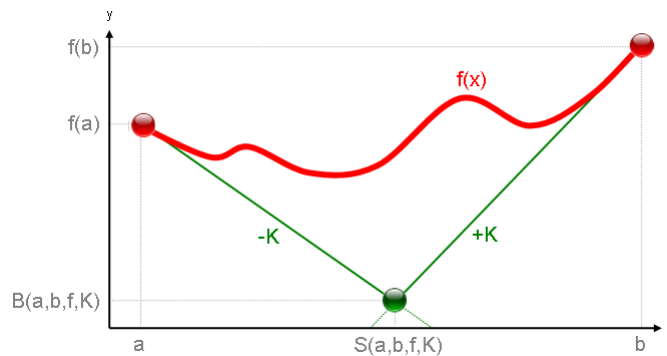


Abbildung 3.3.: untere Schranken einer LIPSCHITZ-stetigen Funktion

#### 3.2.2.2. DiRect-Verfahren<sup>16</sup>

Beim *DiRect-Verfahren* handelt es sich um ein globales Optimierungsverfahren. Es stellt eine Weiterentwicklung der LIPSCHITZ-Optimierung mit dem daraus abgeleiteten bekannten SHUBERT-Algorithmus (siehe [19, 55]) dar, mit dem das globale Minimum einer eindimensionalen LIPSCHITZ-stetigen Funktion bestimmt werden kann.

**Definition<sup>17</sup>:** Eine Funktion  $f : [l, u] \rightarrow \mathbb{R}$  heißt LIPSCHITZ-stetig, wenn eine Konstante<sup>18</sup>  $K \in \mathbb{R}^+$  existiert, so dass gilt:

$$\forall x, \tilde{x} \in [l, u] : |f(x) - f(\tilde{x})| \leq K |x - \tilde{x}| .$$

Für eine LIPSCHITZ-stetige Funktion  $f(x)$  mit  $x \in [a, b] \subseteq [l, u]$  können für alle Funktionswerte die unteren Schranken

$$\begin{aligned} s_1 : f(x) &\geq f(a) - K(x - a), \forall x \in [a, b] \\ s_2 : f(x) &\geq f(b) + K(x - b), \forall x \in [a, b] \end{aligned} \quad (3.1)$$

angeben werden, die zusammengefasst eine V-förmige Beschränkung der LIPSCHITZ-stetigen Funktion darstellen (siehe Abbildung 3.3).

Durch Gleichsetzen der Schranken  $s_1$  und  $s_2$  und passende Umformung<sup>19</sup> ergibt sich für die Spitze  $S$  der Beschränkung

$$S(a, b, f, K) = \frac{a + b}{2} + \frac{f(a) - f(b)}{2K}$$

<sup>16</sup>von Dividing-Rectangles (Teilende Rechtecke), siehe auch [55, 35]

<sup>17</sup>siehe [35]

<sup>18</sup>auch als LIPSCHITZ-Konstante bezeichnet

<sup>19</sup>siehe [55]

### 3. Ausgewählte Optimierungsverfahren

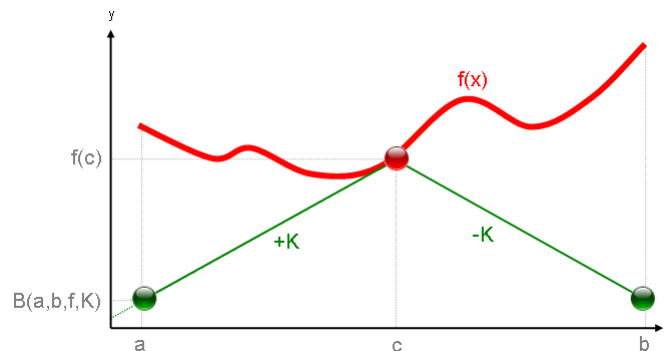


Abbildung 3.4.: untere Schranken des DiRect-Verfahrens

mit

$$B(a, b, f, K) = \frac{f(a) + f(b)}{2} - \frac{K(b-a)}{2} .$$

Beim SHUBERT-Algorithmus beginnt die Optimierung mit der Teilung des Intervalls  $[l, u]$  an der Stelle  $x_0 = S(l, u, f, K)$ . Für die dabei entstandenen Teilintervalle  $[l, x_0]$  und  $[x_0, u]$  wird wiederum  $S$  und  $B$  berechnet. In jedem Iterationsschritt der Optimierung wird das Intervall mit dem kleinsten  $B$  ausgewählt und wiederum geteilt, bis die gewünschte Genauigkeit mit einem vorgegebenen  $\varepsilon$  erreicht ist, so dass für dieses Intervall  $|B - f(S)| < \varepsilon$  gilt. Abhängig von der geschätzten LIPSCHITZ-Konstante  $K$  arbeitet das Verfahren verstärkt lokal (kleines  $K$ ) oder global (großes  $K$ ). Da  $K$  oft nur grob abgeschätzt werden kann und dadurch meist relativ groß ist, wird die globale Suche bevorzugt, wodurch das Verfahren nur langsam konvergiert. Für mehrdimensionale Variablen kann der erste Schritt des Algorithmus recht aufwändig werden, da die Funktionswerte an allen Grenzen der Intervalle<sup>20</sup> berechnet werden müssen.

Beim DiRect-Verfahren wird an Stelle der Grenzen der Mittelpunkt  $c$  des Intervalls betrachtet<sup>21</sup>. Die Schranken (siehe Abbildung 3.4) aus (3.1) werden dann zu

$$\begin{aligned} \tilde{s}_1 : f(x) &\geq f(c) - K(x - c), \forall x \geq c \\ \tilde{s}_2 : f(x) &\geq f(c) + K(x - c), \forall x \leq c \end{aligned} \quad (3.2)$$

mit

$$B(a, b, f, K) = f(c) - \frac{K(b-a)}{2} .$$

Damit  $c$  auch nach der Intervallteilung wieder der Mittelpunkt eines der Teilintervalle ist, wird das Ausgangsintervall in drei äquidistante Teilintervalle zerlegt. Im Unterschied zum SHUBERT-Algorithmus wird für die Auswahl des zu teilenden Intervalls nicht eine Schätzung der LIPSCHITZ-Konstanten verwendet, sondern verschiedene mögliche Konstanten  $K$  betrachtet. Dadurch können unterschiedliche Mittelpunkte gewählt werden. Die zu diesen Mittelpunkten

<sup>20</sup>für  $n$  Dimensionen müssen somit  $2^n$  Funktionswerte berechnet werden.

<sup>21</sup>dadurch muss auch bei mehrdimensionalen Variablen im ersten Schritt nur ein Funktionswert berechnet werden

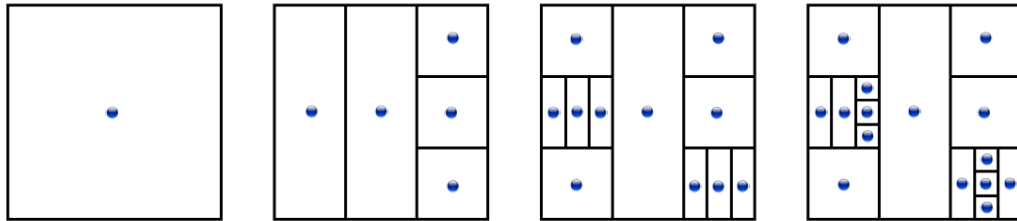


Abbildung 3.5.: Beispiel zum DiRect-Verfahren: mögliche Teilungen des zulässigen Bereichs einer Funktion mit 2 Variablen (Zwischenschritte ausgelassen)

---

**Algorithmus 3.3** DiRect-Verfahren

---

**Eingabe :** OA

Normierung der OA auf Einheits-Hyperwürfel

**while** Abbruchbedingungen nicht erfüllt **do**

- | Finde: potentiell optimale Hyper-Intervalle und deren Mittelpunkt
- | Teile: jeweils entlang der längsten 'Seite' in 3 Teilintervalle

**end**

---

gehörenden Teilintervalle, die einen Ausschnitt der konvexen Hülle bilden, heißen *potentiell optimal*.

Für ein bestimmtes  $K$  wird dann das Teilintervall  $[a, b]$  gewählt, für das  $B$  minimal wird, wobei beachtet werden muss, dass die Intervalle nicht zu klein werden, um zu verhindern, dass sich das Verfahren schnell in einem lokalen Minimum festhängt. Für Funktionen mit mehreren Variablen werden aus den Intervallen Hyper-Intervalle<sup>22</sup>, so dass die Bestimmung der Mittelpunkte entsprechend verallgemeinert wird (siehe [55]).

In Abbildung 3.5 werden am Beispiel einer Funktion mit zwei Variablen mögliche Teilungen des zulässigen Bereichs gezeigt. Der prinzipielle Ablauf des Verfahrens ist in Algorithmus 3.3 dargestellt.

### 3.2.3. Multikriterielle Optimierung durch Überführung in einkriterielle Ersatzoptimierungsaufgaben

Die klassische Herangehensweise zur Lösung MOA besteht darin, eine geeignete einkriterielle Ersatzoptimierungsaufgabe zu formulieren, die dann mit bekannten Verfahren der einkriteriellen Optimierung gelöst werden kann. Die Lösung dieser Ersatzaufgabe stellt dann eine spezielle Lösung der PARETO-Menge dar. Um eine Approximation der PARETO-Menge zu berechnen sind somit mehrere Optimierungen mit verschiedenen Ersatzoptimierungsaufgaben notwendig. Weiterhin haben einige klassische Methoden die Einschränkung, dass möglicherweise nicht alle Teile der PARETO-Menge erreicht werden können. Im Folgenden werden einige Methoden

---

<sup>22</sup>z.B. für 2 Variablen werden es Rechtecke

### 3. Ausgewählte Optimierungsverfahren

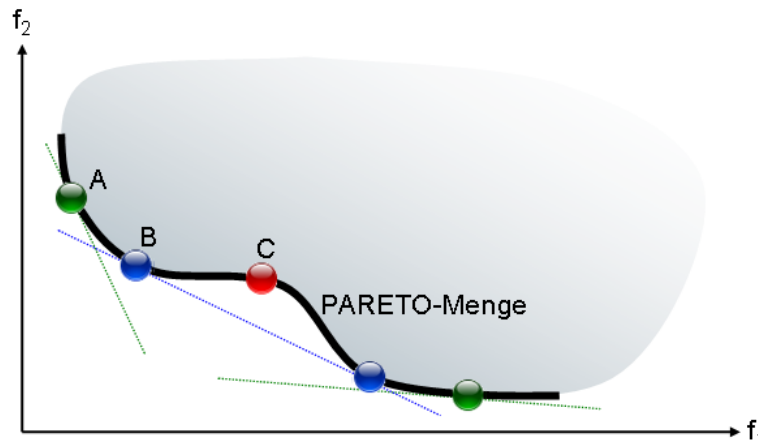


Abbildung 3.6.: Gewichtete-Summen-Methode (Beispiel mit 2 Zielen)

- A ... erreichbarer Punkt → auf konvexem Bereich der PARETO-Menge
- B ... nicht eindeutig erreichbarer Punkt → am Rand des konvexen Bereichs der PARETO-Menge
- C ... nicht erreichbarer Punkt → nicht auf konvexem Bereich der PARETO-Menge

betrachtet<sup>23</sup>, die für die praktische Anwendung von Bedeutung sind.

#### 3.2.3.1. Gewichtete-Summen-Methode

Bei dieser Methode wird die MOA (siehe 2.1) in eine EOA transformiert, indem die einzelnen Zielfunktionen jeweils mit einem Gewicht multipliziert und dann aufsummiert werden. Dabei ist es aber erforderlich, dass zuvor die Zielfunktionen  $f_m(\mathbf{x})$ ,  $m = 1, 2, \dots, M$  normiert werden, da deren Wertebereiche sehr verschieden sein können<sup>24</sup>. Die einkriterielle Ersatzaufgabe lautet dann

$$\left\{ \begin{array}{l} \sum_{m=1}^M w_m f_m(\mathbf{x}) \rightarrow \min \\ g_j(\mathbf{x}) \geq 0 \quad j = 1, 2, \dots, J \\ x_i^{(U)} \leq x_i \leq x_i^{(O)} \quad i = 1, 2, \dots, N \\ \mathbf{x} \in \mathbb{R}^N \\ \sum_{m=1}^M w_m = 1, w_m \in [0, 1] \end{array} \right. \quad (3.3)$$

Bei der Verwendung dieser Methode muss beachtet werden, dass nur konvexe Bereiche der PARETO-Menge erreicht werden können (siehe Abbildung 3.6). Die Anwendung dieser Methode ist sehr anschaulich und einfach, indem durch die Wahl der Gewichte  $w_m$  mit  $m =$

<sup>23</sup>siehe auch [12, 53, 58]

<sup>24</sup>In der praktischen Anwendung stellt dies meist ein Problem dar, da vor der Optimierung die Wertebereiche nicht bekannt sind und deshalb nur geschätzt werden können bzw. mit erheblichem Aufwand ermittelt werden müssen.



### 3.2. Deterministische Verfahren und deren multikriterielle Erweiterungen

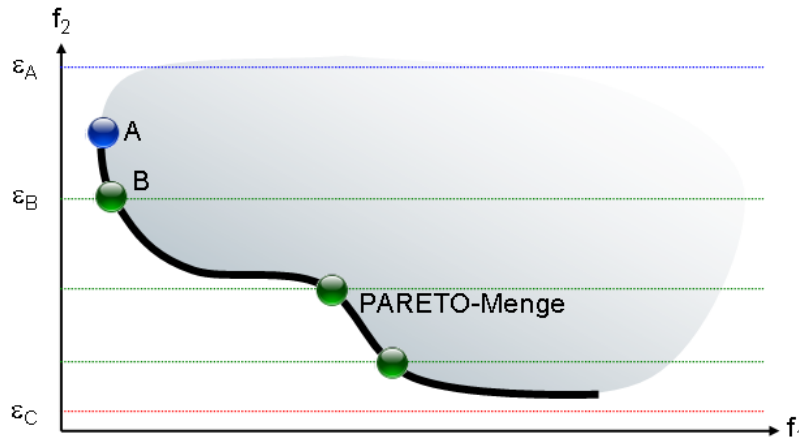


Abbildung 3.7.:  $\varepsilon$ -Constraint-Methode (Beispiel mit 2 Zielen)

- $\varepsilon_A$  ... keine Einschränkung des Zielgebietes
- $\varepsilon_B$  ... Einschränkung zur Wahl des Punktes B
- $\varepsilon_C$  ... die zulässige Menge ist leer  $\rightarrow$  keine Lösung

1, 2, ...,  $M$  die Wichtigkeit der einzelnen Ziele festgelegt wird.

#### 3.2.3.2. $\varepsilon$ -Constraint-Methode

Bei der  $\varepsilon$ -Constraint-Methode<sup>25</sup> wird von den Zielfunktionen eine Funktion  $f_\mu(\mathbf{x})$  mit  $\mu \in \{1, 2, \dots, M\}$  als Zielfunktion der Ersatzaufgabe gewählt. Die restlichen Funktionen werden in Nebenbedingungen mit den oberen Schranken  $\varepsilon_m$ ,  $m = 1, 2, \dots, M \wedge m \neq \mu$  umgewandelt:

$$\begin{cases} f_\mu(\mathbf{x}) \rightarrow \min & \mu \in \{1, 2, \dots, M\} \\ f_m(\mathbf{x}) \leq \varepsilon_m & m = 1, 2, \dots, M \wedge m \neq \mu \\ g_j(\mathbf{x}) \geq 0 & j = 1, 2, \dots, J \\ x_i^{(U)} \leq x_i \leq x_i^{(O)} & i = 1, 2, \dots, N \\ \mathbf{x} \in \mathbb{R}^N \end{cases} \quad (3.4)$$

Während die Auswahl von  $\mu$  in vielen Fällen gut möglich ist, stellt die Festlegung der Schranken  $\varepsilon_m$  für die praktische Anwendung meist eine Herausforderung dar, da die Wertebereiche der Zielfunktionen bekannt sein müssen, um sinnvolle Schranken festlegen zu können. Mit der  $\varepsilon$ -Constraint-Methode können sowohl konvexe als auch konkave Bereiche der PARETO-Menge erreicht werden.

In Abbildung 3.7 wird die Methode an einem Beispiel mit zwei zu minimierenden Zielfunktionen visualisiert.

<sup>25</sup>auch Beschränkungsmethode genannt

### 3. Ausgewählte Optimierungsverfahren

#### 3.2.3.3. Gewichtete-Metrik-Methode

Bei dieser Methode wird der gewichtete Abstand der Lösung  $f(\mathbf{x})$  vom Idealpunkt  $\mathbf{z}^*$  verwendet, um die einzelnen Zielfunktionen  $f_m(\mathbf{x})$ ,  $m = 1, 2, \dots, M$  zu einer Zielfunktion zu kombinieren. Analog zur Gewichteten-Summen-Methode werden auch bei dieser Methode nicht negative Gewichte  $w_m$ ,  $m = 1, 2, \dots, M$  verwendet. Mit  $p \in [1, \infty)$  erhält man die Ersatzoptimierungsaufgabe

$$\left\{ \begin{array}{l} \left( \sum_{m=1}^M w_m |f_m(\mathbf{x}) - z_m^*|^p \right)^{\frac{1}{p}} \rightarrow \min \\ g_j(\mathbf{x}) \geq 0 \quad j = 1, 2, \dots, J \\ x_i^{(U)} \leq x_i \leq x_i^{(O)} \quad i = 1, 2, \dots, N \\ \mathbf{x} \in \mathbb{R}^N \\ \sum_{m=1}^M w_m = 1, w_m \in [0, 1] \end{array} \right. \quad (3.5)$$

Die Wahl des Parameters  $p$  der Metrik hat sehr großen Einfluss auf die zu erreichende Lösung. Nur für  $p = \infty$ <sup>26</sup> kann jede Lösung der PARETO-Front erreicht werden. Durch geeignete Anpassungen<sup>27</sup>, wie z.B. die Ausführung einer zusätzlichen Drehung oder der dynamischen Anpassung des Idealpunktes, kann die Effektivität dieser Methode erhöht werden.

Bei der BENSON-Methode handelt es sich um eine Abwandlung der Gewichtete-Metrik-Methode. Anstelle der Berechnung des Minimalen Abstandes vom Idealpunkt wird hier der maximale Abstand eines zufällig gewählten, nicht PARETO-optimalen, aber zulässigen Punktes verwendet.

#### 3.2.3.4. Nutzenfunktionsmethode

Für die Nutzenfunktionsmethode wird eine Funktion  $U : \mathbb{R}^M \rightarrow \mathbb{R}$  mit

$$U = U(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x}))$$

definiert, die den Nutzen einer Lösung bewertet. Die Nutzenfunktion  $U$  muss dazu auf der gesamten zulässigen Menge  $S$  definiert sein. Für eine streng monoton fallende Nutzenfunktion liefert das Maximum dann eine PARETO-optimale<sup>28</sup> Lösung. Damit erhält man die zu

---

<sup>26</sup>Tschebyscheff-Norm

<sup>27</sup>siehe [12]

<sup>28</sup>siehe [12], Theorem von MIETTINEN

### 3.2. Deterministische Verfahren und deren multikriterielle Erweiterungen

maximierende Ersatzoptimierungsaufgabe

$$\begin{cases} U(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})) \rightarrow \max \\ g_j(\mathbf{x}) \geq 0 & j = 1, 2, \dots, J \\ x_i^{(U)} \leq x_i \leq x_i^{(O)} & i = 1, 2, \dots, N \\ \mathbf{x} \in \mathbb{R}^N \end{cases} \quad (3.6)$$

In Abhängigkeit von der Nutzenfunktion  $U$  ist es auch bei dieser Methode<sup>29</sup> möglich, dass nicht alle Bereiche der PARETO-Front erreicht werden können. Da die Wahl einer geeigneten Nutzenfunktion sehr problemabhängig ist, ist es nicht möglich, eine allgemein 'beste' Nutzenfunktion anzugeben.

#### 3.2.3.5. Pareto-Scan

Beim PARETO-Scan handelt es sich nicht um eine eigenständige Methode, sondern um einen einfachen Ansatz, mit klassischen Methoden nicht nur eine, sondern mehrere PARETO-optimale Lösung zu finden. Dazu werden verschiedene Optimierungsläufe nacheinander mit systematisch veränderten Verfahrensparametern (z.B. Gewichte, Schranken, Normierungskoeffizienten usw.) durchgeführt, wobei als Ausgangspunkt jeweils die Lösung der vorangegangenen Optimierung verwendet wird.

Das bedeutet zum Beispiel beim Einsatz der Gewichteten-Summe-Methode, dass eine vorgegebene Anzahl von möglichen Kombinationen der Gewichte nacheinander berechnet werden. Sind deren Unterschiede zwischen zwei Teiloptimierungen relativ klein, konvergiert die darauf folgende Optimierung in der Regel schnell<sup>30</sup>.

#### 3.2.3.6. Zusammenfassung

Die klassischen Methoden ermöglichen den Einsatz bekannter einkriterieller Optimierungsverfahren zur Lösung von MOA durch die Transformation der multikriteriellen Aufgabe in eine einkriterielle Ersatzaufgabe. Mit der Einschränkungen, dass manche Methoden nicht die ganze PARETO-Menge erreichen können und dass stets nur eine einzelne PARETO-optimale Lösung gefunden wird<sup>31</sup>, sind diese Methoden in der praktischen Anwendung einsetzbar, wenn zum Beispiel der gesuchte Kompromiss bekannt ist oder eine Rasterung mittels PARETO-Scan ausreichend ist. Aber insbesondere dann, wenn im Postprozess verschiedene Kompromisse betrachtet werden müssen, ist der Einsatz multikriterieller Verfahren vorteilhaft, die nicht nach einer optimalen Lösung, sondern der PARETO-Menge suchen.

<sup>29</sup>die Gewichtete-Summe-Methode und die Gewichtete-Metrik-Methode sind Spezialfälle der Nutzenfunktionsmethode

<sup>30</sup>d.h. die Kompromisse liegen dicht nebeneinander auf der PARETO-Front

<sup>31</sup>beim Einsatz von PARETO-Scan werden einige wenige Lösungen ermittelt

## 3.3. Stochastische Verfahren

Die *stochastischen Optimierungsverfahren*, die in dieser Arbeit betrachtet werden, sind Algorithmen, bei denen man sich durch zufällige Änderungen der Optimierungsvariablen und anschließende Bewertung und Auswahl an das Optimum herantastet<sup>32</sup>. Die Art der zufälligen Änderungen und der Auswahl werden durch die gewählte Methode oder Heuristik bestimmt. Im einfachsten Fall, der *Monte-Carlo-Methode* (Abschnitt 3.3.2), werden die Werte der Variablen einfach ausgewürfelt und das Optimum durch Vergleich der Werte der Zielfunktion<sup>33</sup> ausgewählt. Andere stochastische Verfahren, die *naturinspirierten Verfahren*, bilden als Heuristik Vorgänge oder Verhaltensweisen aus der Natur ab. Komplexe stochastische Verfahren, zu denen die *adaptiven Ersatzmodellverfahren* zählen, erstellen ein statistisches Modell der Problemstellung und leiten daraus Vorhersagen für die nächsten Varianten ab.

Die Art der Heuristik sollte passend zur gegebenen Aufgabenstellung gewählt werden und ist ausschlaggebend dafür, wie viele Varianten berechnet werden müssen und bestimmt maßgeblich den Rechenaufwand des Optimierungsverfahrens.

Eine wichtige Eigenschaft stochastischer Verfahren ist, dass das Erreichen des tatsächlich Optimums in der Regel weder gesichert noch nachweisbar ist. Für die praktische Anwendung, zum Beispiel bei technischen Aufgabenstellungen, ist dies aber meist kein Problem, da eine signifikante Verbesserung auch dann wertvoll ist, wenn es sich dabei nicht um das globale Optimum handelt.

### 3.3.1. Zufallszahlen

Da man bei stochastischen Verfahren die Optimierungsvariablen als Zufallsgrößen betrachtet, werden Zufallszahlen in vielfältiger Form benötigt. In numerischen Algorithmen werden meist Pseudozufallszahlen verwendet, die durch arithmetische Generatoren erzeugt werden. Bei Zahlenfolgen, die auf diese Weise entstehen, handelt es sich somit im strengen Sinne nicht um Zufallszahlen, da sie zum einen periodisch sind<sup>34</sup> und zum anderen bei Kenntnis der Erzeugungsvorschrift weitere Glieder der Folge vorhersagbar sind. Für die praktische Anwendung können Pseudozufallszahlen aber sehr gut eingesetzt werden. Dabei kann es aber auch als Vorteil dieser Generatoren angesehen werden, dass bei definierten Startbedingungen exakt die gleiche Zahlenfolge erzeugt wird und somit reproduzierbare Ergebnisse produziert werden können<sup>35</sup>.

Brauchbare Pseudozufallszahlen-Generatoren für gleichverteilte Zufallszahlen müssen den folgenden Forderungen genügen (siehe auch [23, 37, 38]):

<sup>32</sup>es existieren auch stochastische Verfahren, die ohne Bewertung arbeiten

<sup>33</sup>bei der multikriteriellen Optimierung wird die PARETO-Menge bestimmt und die Auswahl auf Basis einer Fitnessfunktion durchgeführt

<sup>34</sup>wenn auch mit zum Teil sehr langen Perioden

<sup>35</sup>auf diese Weise sind zum Beispiel reproduzierbare und somit vergleichbare Tests möglich

**Lange Perioden:** Die Periode eines Generators muss lang genug sein, so dass sich die benötigte Folge von Zufallszahlen innerhalb einer Periode befindet.

**Uniformität:** Die Gleichverteilung der Zufallszahlen aller erzeugten Folgen muss gewährleistet sein.

**Unabhängigkeit:** Zwischen den Zufallszahlen innerhalb jeder erzeugten Folge dürfen keine Korrelationen bestehen.

Der Algorithmus *Mersenne-Twister-19937*, der von MAKOTO MATSUMOTO und TAKUJI NISHIMURA entwickelt wurde<sup>36</sup>, erfüllt diese Kriterien und ist mit einer Periode von  $2^{19937} - 1$  und einer hochgradigen Gleichverteilung<sup>37</sup> sehr gut für das Erzeugen von Zufallszahlen in stochastischen Verfahren einsetzbar.

### 3.3.2. Monte-Carlo-Methode

Das einfachste stochastische Suchverfahren, die *Monte-Carlo-Methode* oder *Monte-Carlo-Optimierung*<sup>38</sup>, besteht darin, die Werte der Optimierungsvariablen im zulässigen Bereich auszuwürfeln und anschließend die beste Lösung auszuwählen<sup>39</sup>. Dabei werden die Optimierungsvariablen als gleichverteilte Zufallsgrößen in den jeweils gültigen Intervallen betrachtet.

Die MCO ist für beliebige Problemstellungen anwendbar, arbeitet aber nicht effektiv, da keinerlei Informationen aus vorangegangenen Schritten ausgenutzt werden. Aussagen darüber, ob mit einer begrenzten Anzahl von Varianten das Optimum erreicht oder die PARETO-Menge mit ausreichender Genauigkeit gefunden wurde, sind nicht möglich. Da alle Varianten für sich unabhängig betrachtet werden, ist insbesondere bei kleinen Anzahlen von Varianten nicht gewährleistet, dass der mögliche Wertebereich gleichmäßig durchsucht wird.

Trotz seiner allgemeinen Gültigkeit und Einfachheit ist das Verfahren ungeeignet für den praktischen Einsatz. Es wird aber als Referenz-Verfahren zur Beurteilung der Effektivität anderer Optimierungsverfahren verwendet.

Ein effektives Verfahren muss im statistischen Mittel

- mit gleichem Aufwand bessere oder
- mit weniger Aufwand gleich gute

Lösungen finden als die MCO.

Durch einfache Erweiterungen können aus der MCO weitere Optimierungsmethoden abgeleitet werden, die trotzdem eine sehr simple Struktur haben, aber Eigenschaften aufweisen,

---

<sup>36</sup>siehe [38]

<sup>37</sup>bewiesene Gleichverteilung bis zur Dimension 623, siehe [38]

<sup>38</sup>kurz: MCO

<sup>39</sup>Analog lässt die Methode sich durch Bestimmen der PARETO-Menge auch für multikriterielle Problemstellungen anwenden.

### 3. Ausgewählte Optimierungsverfahren

---

**Algorithmus 3.4** Verfahren: Monte-Carlo-Optimierung

---

**Eingabe** : OA

```
while Abbruchbedingungen nicht erfüllt do  
  | erzeuge und berechne zufällige Varianten  
end  
bewerte alle Varianten und wähle Beste aus
```

---

wodurch sie effektiver arbeiten. Beispielhaft werden zwei Methoden angeführt, die zum Teil auch Anwendung in der Praxis finden:

**Latin-Hypercube-Methode:** Bei der LHM werden die Variablen der Varianten nicht durch unabhängige, gleichverteilte Zufallszahlen belegt, sondern mittels Latin-Hypercube-Sampling (siehe [44, 58]) ermittelt. Auf diese Weise kann erreicht werden, dass zumindest der gesamte Wertebereich gleichmäßig mit Varianten abgedeckt wird. Alle anderen Eigenschaften der MCO bleiben allerdings unverändert. Diese Methode wird oftmals als Initialisierung verwendet, wenn keine weiteren Informationen vorhanden sind und ist deutlich günstiger, als eine unabhängige zufällige Initialisierung.

**Stochastic Design Improvement:** Das Verfahren SDA ist auch unter der Bezeichnung Bergsteiger-Algorithmus<sup>40</sup> bekannt. Dabei wird in jedem Schritt des Verfahrens ein MCO oder LHM für eine vorgegebene Anzahl  $n$  ausgeführt und die beste Lösung gesucht. Bevor der nächste Schritt ausgeführt wird, werden die Intervalle der Variablen dergestalt neu festgelegt, dass sie nur noch aus der Nachbarschaft der besten Lösung bestehen. Die Art der Bestimmung der neuen Intervalle hat dabei großen Einfluss auf den Verlauf der Optimierung. Werden die Intervalle nur sehr langsam verkleinert, konvergiert auch die Optimierung nur sehr langsam, werden sie allerdings zu stark verkleinert, konvergiert die Optimierung möglicherweise sehr schnell im nächst besten lokalen Extremum.

Der Algorithmus (siehe Algorithmus 3.5) ist gut für einkriterielle Aufgaben einsetzbar und besonders dann, wenn die Zielfunktion relativ glatt ist<sup>41</sup>, so dass die Intervalle in jedem Iterationsschritt relativ stark verkleinert werden können, kann eine schnelle Konvergenz erreicht werden.

Soll das Verfahren für multikriterielle Aufgabenstellungen eingesetzt werden, müssen die neuen Intervalle auf der Basis der PARETO-Menge bestimmt werden. Dadurch ist es nicht möglich, eine eindeutige Variante festzulegen, von der ausgehend die neuen Intervalle festgelegt werden und der nächste Iterationsschritt ausgeführt wird. Als mögliche Strategie können mehrere Elemente der PARETO-Menge als separate Ausgangspunkte zur Fortsetzung gewählt werden<sup>42</sup>. Bei vielen solchen Ausgangspunkten sinkt allerdings die Effektivität des Verfahrens drastisch

---

<sup>40</sup>engl. Hillclimber-Algorithm

<sup>41</sup>d.h. wenige lokale Extrema besitzt

<sup>42</sup>zum Beispiel gut verteilt über die ganze Menge oder Varianten mit hoher Fitness

---

**Algorithmus 3.5** Verfahren: Stochastic Design Appovement

---

**Eingabe** : OA

```

while Abbruchbedingungen nicht erfüllt do
  | Neue Varianten mittels LHS erzeugen und berechnen
  | Bestimme die beste Variante
  | schrumpfe die Intervalle mit der besten Variante als Zentrum
end

```

---

ab, werden dagegen nur sehr wenige verwendet, wird die Approximation der PARETO-Front nicht gleichmäßig verteilt sein.

**3.3.3. Naturinspirierte Verfahren**

Naturinspirierte Verfahren sind heuristische Methoden, die Vorgänge, meist aus der Biologie oder Physik, simulieren oder Verhaltensweisen von Tieren nachbilden. Auch wenn der Zufall weiterhin eine entscheidende Rolle spielt, ist die verwendete Heuristik ausschlaggebend für die Eigenschaften des Verfahrens. Typische Vertreter für Verfahren, die physikalische oder biologische Vorgänge simulieren, sind

- *Evolutionäre Verfahren*<sup>43</sup>, die die Gesetze der Genetik<sup>44</sup> abbilden und
- das Verfahren *Simulated Annealing*, bei dem das Abkühlen einer Schmelze simuliert wird.

Zu den Verfahren, die die Verhaltensweise von Tieren bzw. Tierschwärmen nachbilden und als Basis für die Optimierung nutzen, zählen

- die *Partikelschwarm Verfahren*, bei der das Schwarmverhalten von Tierschwärmen<sup>45</sup> simuliert wird und
- *Ameisen*-<sup>46</sup> bzw. *Ameisenkolonie Verfahren*, die das Verhalten von Ameisen auf der Futtersuche nachbilden.

Alle naturinspirierten Verfahren haben gemeinsam, dass natürliche Prozesse, die zur schrittweisen Verbesserung oder zu Effekten wie der Ausbildung geordneter Strukturen führen, auf mathematische Algorithmen übertragen werden. Dabei sind die natürlichen Vorgänge zum Teil sehr exakt algorithmisch umgesetzt (z.B. Evolutionäre Verfahren oder Partikelschwarmverfahren) oder es werden noch weitere Schritte eingefügt oder abgewandelt (z.B. Modellbasierte Evolutionäre Algorithmen oder Ameisenkolonieverfahren).

---

<sup>43</sup>dazu zählen u.a. auch *Genetische Algorithmen*

<sup>44</sup>MENDELSche Gesetze

<sup>45</sup>zum Beispiel Fischenschwärmen

<sup>46</sup>eine Abwandlungen des Verfahrens ist z.B. das Bienenverfahren

### 3. Ausgewählte Optimierungsverfahren

---

#### Algorithmus 3.6 Evolutionäre Verfahren

---

**Eingabe** : OA

Startpopulation mittels LHS o.ä. erzeugen

**while** *Abbruchbedingungen nicht erfüllt* **do**

    Selektion

    Variation (z.B. Kreuzung, Mutation)

    Aufbau einer neuen Population

**end**

---

#### 3.3.3.1. Evolutionäre Verfahren

Evolutionäre Verfahren, zu denen Evolutionsstrategien und Genetische Algorithmen gehören, sind heuristische Methoden, die die Vorgänge der Vererbung, die aus der Biologie bekannt sind<sup>47</sup>, nachbilden und auf mathematische Probleme übertragen. Wie sich in der Natur beobachten lässt, setzen sich durch wiederholte Selektion, Mischung (Kreuzung) und zufällige Veränderungen (Mutation) von Eigenschaften diejenigen Individuen<sup>48</sup> durch, die durch ihre Eigenschaften besonders gut an ihren Lebensraum angepasst sind, während Individuen mit unpassenden Eigenschaften nach und nach verschwinden. Die Übertragung in einen mathematischen Algorithmus kann hierbei sehr stark an den biologischen Prozess angelehnt sein oder auch auf komplexeren Modellen beruhen<sup>49</sup>. Die Idee wurde in den 1960er Jahren erstmals von JOHN A. HOLLAND und zeitgleich von INGO RECHENBERG vorgestellt und seitdem intensiv untersucht und weiterentwickelt.

Das Grundprinzip ist dabei aber immer ähnlich. Ausgehend von einer Startpopulation<sup>50</sup>, die mit zufällig erzeugten oder schon bekannten Individuen aufgebaut ist, wird eine Selektion durchgeführt, mit der die besten Individuen herausgefiltert bzw. die Individuen bezüglich ihrer Fitness sortiert werden. Daran schließt sich eine Variation der Individuen, deren Berechnung und Neuordnung in der Nachkommen-Population an. Dieser Zyklus wird für jede Generation<sup>51</sup> immer wieder durchlaufen (siehe Algorithmus 3.6).

Zur Variation der Individuen werden Operatoren definiert, die ebenfalls von Abläufen, die aus der Genetik bekannt sind, abgeleitet werden.

**Kreuzung:** Bei der Kreuzung werden Eigenschaften zweier Individuen miteinander kombiniert. Dies geschieht bei diskreten Variablen z.B. durch Mischen von Teilen der Binärstrings, die diese Variable beschreiben. Bei stetigen Variablen wird die neue Ausprägung z.B. durch eine Linearkombination der Ausgangsindividuen erzeugt. Ziel der Kreuzung ist es, Eigenschaften mehrerer Individuen zu kombinieren und im Zusammenspiel mit der Selektion die Ausprägungen

---

<sup>47</sup>Genetik, MENDELsche Gesetze

<sup>48</sup>Elementen der zulässigen Menge  $S$

<sup>49</sup>siehe in [3, 43, 58]

<sup>50</sup>Dabei stellt eine Population eine Menge von Individuen dar.

<sup>51</sup>= Iteration



zu verstärken, die zu einer guten Fitness führen.

**Mutation:** Bei der Mutation wird das gewählte Individuum bzw. ein Teil von ihm zufällig verändert. Ziel der Mutation ist es, durch diese zufälligen Änderungen neue Informationen zu gewinnen und die Lösungsvielfalt zu erhöhen, indem so neue Wertebereiche betrachtet werden, die durch Kreuzung nicht erreicht würden. Weiterhin wird dadurch verhindert, dass der Algorithmus zu schnell konvergiert und damit vielleicht in einem lokalen Extrempunkt hängen bleibt.

Zur Bewertung der besten Individuen kann bei einkriteriellen Optimierungsaufgaben der Funktionswert der Zielfunktion direkt verwendet werden. Sind mehrere Ziele vorhanden, wird zur Bewertung ein Fitness-Wert gebildet, mit dem die Wichtigkeit dieser Lösung für eine gute und gleichmäßige Approximation der PARETO-Menge beurteilt werden kann. Eine mögliche Methode ist die *Methode der nichtdominierten Sortierung*<sup>52</sup> (NSM), die in [12] vorgestellt und inzwischen in vielen multikriteriellen Optimierungsalgorithmen (z.B. NSGA-II) eingesetzt wird und an deren Beispiel die Fitnessbewertung demonstriert werden soll. Darüber hinaus gibt es aber auch eine Vielzahl weiterer moderner Verfahren<sup>53</sup> (z.B. SPEA2, SMS-EMOA, MSOPS), die andere Methoden zur Fitness-Bewertung verwenden.

Bei der NSM wird die Population in Ränge hinsichtlich ihre Nähe zur PARETO-Front eingeteilt und auf Basis der Strategien *Adaptive-Sharing* und *Dynamic-Crowding* gut verteilte Individuen ausgewählt.

**Ranking:** Jedem Individuum  $\mathbf{x}^{(j)}$  einer Population wird ein Rang  $R(\mathbf{x}^{(j)})$  zugewiesen. Alle Individuen, die zur PARETO-Menge gehören, erhalten den Rang eins. Der nächste Rang kann leicht ermittelt werden, indem alle PARETO-optimalen Individuen entfernt werden und die PARETO-Menge der restlichen Punkte bestimmt wird. Dies wird fortgesetzt, bis allen Individuen ein Rang zugewiesen wurde.

Da für die Erzeugung der neuen Population besonders solche Individuen gesucht werden, die die PARETO-Menge möglichst gut approximieren, werden Individuen mit kleinem Rang bevorzugt.

**Dynamic Crowding:** Die *Crowding-Distance*<sup>54</sup> ist ein Maß für die Dichte der Punkte der PARETO-Front. Sie ist für ein Individuum Null, wenn dieses mehrmals in der Population auftritt. Für die Randpunkte der PARETO-Front  $P$  wird die  $CD$  auf  $\infty$  gesetzt. Zur Berechnung der  $CD$  werden die Elemente der PARETO-Front zunächst lexikographisch bezüglich der einzelnen Zielfunktionswerte geordnet. Die Crowding-Distance eines Individuums  $\mathbf{x}^{(j)} \in Z$  ist

---

<sup>52</sup>non-dominated sorting method

<sup>53</sup>siehe auch [51]

<sup>54</sup>kurz  $CD$

### 3. Ausgewählte Optimierungsverfahren

dann definiert<sup>55</sup> durch:

$$CD(\mathbf{x}^{(j)}) = \begin{cases} \sum_{m=1}^M \frac{\delta_m(\mathbf{x}^{(j+1)}, \mathbf{x}^{(j-1)}) \delta_m(\mathbf{x}^{(j)}, \mathbf{x}^{(j-1)}) \delta_m(\mathbf{x}^{(j+1)}, \mathbf{x}^{(j)})}{(f_m^{max} - f_m^{min})^3} & j = 2, \dots, |P| - 1 \\ \infty & \text{sonst} \end{cases}$$

mit

$$\delta_m(\mathbf{x}^{(a)}, \mathbf{x}^{(b)}) = f_m(\mathbf{x}^{(a)}) - f_m(\mathbf{x}^{(b)}).$$

Die Werte  $f_m^{min}$  und  $f_m^{max}$  sind der kleinste bzw. größte Wert der  $m$ -ten Zielfunktion. Gesucht werden die Individuen  $\mathbf{x}^{(j)}$ , bei denen die  $CD(\mathbf{x}^{(j)})$  groß ist. Damit werden solche Punkte gewählt, die möglichst gleichmäßig verteilt auf der PARETO-Front liegen.

**Adaptive Sharing:** Mit der *Adaptive-Sharing-Methode* wird die Verteilung der Individuen einer Population anhand der Anzahl der dominierenden Individuen geschätzt. Dabei ist  $N_{DOM}(\mathbf{x}^{(j)})$ ,  $j = 1, \dots, |P|$  die Anzahl der Individuen, die das  $j$ -te Individuum  $\mathbf{x}^{(j)}$  dominieren. Mit der *Sharing Intensity*

$$SI(\mathbf{x}^{(j)}) = 1 - \frac{1}{1 + N_{DOM}(\mathbf{x}^{(j)})}, j = 1, \dots, |P|$$

wird die Entfernung eines Individuums von der PARETO-Front bewertet. Nicht dominierte Individuen haben immer die Sharing Intensity Null.

Bei der Auswahl von Individuen, die nicht zur PARETO-Front gehören, werden solche bevorzugt, die eine kleine  $SI$  haben.

**Fitness:** Die Fitness  $F(\mathbf{x}^{(j)})$  eines Individuums wird dann aus dem Rang  $R(\mathbf{x}^{(j)})$  und entweder der Crowding-Distance  $CD(\mathbf{x}^{(j)})$  für PARETO-optimale Individuen oder der Sharing Intensity  $SI(\mathbf{x}^{(j)})$  berechnet:

$$F(\mathbf{x}^{(j)}) = \begin{cases} 1 - \frac{1}{1 + CD(\mathbf{x}^{(j)})} & \text{für } R(\mathbf{x}^{(j)}) = 1 \\ R(\mathbf{x}^{(j)}) + \gamma SI(\mathbf{x}^{(j)}) & \text{sonst} \end{cases}.$$

Da die Randpunkte der PARETO-Front  $P$  eine Crowding-Distance von  $\infty$  haben, wird  $\frac{1}{1 + CD(\mathbf{x}^{(j)})}$  Null gesetzt. Mit dem Parameter  $\gamma$  kann der Einfluss<sup>56</sup> der Sharing Intensity gesteuert werden.

Die Fitness  $F(\mathbf{x}^{(j)})$  eines Individuums ist klein, wenn es dicht an der PARETO-Front liegt und zu einer gut verteilten Approximation der PARETO-Menge gehört.

<sup>55</sup>In der Literatur sind verschiedene Definitionen der  $CD$  angegeben. Die hier angegebene erweiterte Formulierung hat sich in Untersuchungen in den meisten Fällen als treffsicherer erwiesen.

<sup>56</sup>Der Wert von  $\gamma$  sollte nicht zu groß gewählt werden (siehe [12, 58]).

Moderne Weiterentwicklungen dieser Algorithmen verfolgen meist das Ziel, durch Modellierung und erweiterte Analyse der schon berechneten Varianten, mehr Informationen über die gegebene Optimierungsaufgabe zu erhalten und somit die Konvergenzgeschwindigkeit zu erhöhen. Das ist z.B. dann von Bedeutung, wenn viele Optimierungsvariablen verwendet werden. Beispielsweise werden bei dem Algorithmus MrBOA (siehe [3, 58]) die klassischen Variationsoperatoren Kreuzung und Mutation durch das Erlernen von Abhängigkeiten zwischen den Variablen und deren Abbildung durch ein Wahrscheinlichkeitsmodell ersetzt<sup>57</sup>. Ziel der Modellbildung ist es, die linearen und nichtlinearen Verbindungen der Optimierungsvariablen zu erlernen und durch eine geschätzte Wahrscheinlichkeitsverteilung der Variablen zu beschreiben. Da die Elemente der zulässigen Menge  $S$  den Individuen entsprechen, die der PARETO-Front  $P$  am nächsten liegen, repräsentiert die Verteilung der zugehörigen Variablen die Wahrscheinlichkeit für einen bestimmten Variablenwert, Zielfunktionswerte nahe der PARETO-Front  $P$  zu erzeugen.

#### 3.3.3.2. Partikelschwarm Verfahren<sup>58</sup>

Die Schwarmoptimierung ist ebenfalls eine populationsbasierende stochastische Optimierungsstrategie, die auf Beobachtungen des Schwarmverhaltens von Tieren<sup>59</sup> basiert. Solche Tier Schwärme bestehen aus einer großen Anzahl von Einzelindividuen, die für sich betrachtet nur eine relativ geringe Intelligenz besitzen. Die Individuen des Schwarms stehen miteinander, insbesondere mit ihren direkten Nachbarn, in einem engen Kontakt, so dass sie die Informationen, wo sich ein Optimum befindet, austauschen können und sich so der ganze Schwarm in die Gebiete bewegt, wo die Erfolgsaussichten gut sind.

Die Partikelschwarm-Optimierung (PSO) wurde von JAMES KENNEDY und RUSSELL EBERHART im Jahre 1995 eingeführt<sup>60</sup> und wurde besonders in den letzten Jahren intensiv untersucht und erfolgreich angewendet.

Das zugrundeliegende Prinzip der PSO ist einfach. Die einzelnen Partikel<sup>61</sup> müssen dabei nur wenigen, einfachen Regeln gehorchen:

1. Ausrichtung: Jedes Partikel bewegt sich in die ähnliche Richtung wie seine Nachbarn.
2. Anziehung: Sind die Nachbarn des Partikels weit entfernt, so nähert sich das Partikel diesen an.
3. Abstoßung: Kommen sich einzelne Partikel zu nahe, bewegen sie sich zur Vermeidung von Kollisionen voneinander weg.

<sup>57</sup>Probabilistische Genetische Algorithmen

<sup>58</sup>siehe [49]

<sup>59</sup>zum Beispiel von Fisch- oder Vogelschwärmen

<sup>60</sup>siehe [26]

<sup>61</sup>die Individuen des Schwarms werden meist als Partikel bezeichnet

### 3. Ausgewählte Optimierungsverfahren

---

**Algorithmus 3.7** Partikelschwarm Verfahren

---

**Eingabe** : OA

Startpositionen und -geschwindigkeiten des Schwarms mittels LHS o.ä. initialisieren

```
while Abbruchbedingungen nicht erfüllt do
  Bestimmung der besten Position im Schwarm
  forall Partikel do
    Update der Geschwindigkeit
    Berechnung der neuen Position (Funktionsauswertung)
  end
end
```

---

4. Jedes Partikel strebt einen Rastplatz an, will aber auch im Schwarm verbleiben

Die Individuen des Schwarms, die mögliche Lösungen der OA darstellen, stellen kollisionsfreie Partikel dar und sind auf der Suche nach dem besten Rastplatz<sup>62</sup>, der durch den Parametervektor der erwarteten optimalen Lösung definiert ist. Für jedes Partikel  $s \in \mathcal{S}$  des Schwarmes  $\mathcal{S}$  ist seine Position  $\mathbf{p}^s = (p_1^s, \dots, p_N^s)^T$  und seine Geschwindigkeit  $\mathbf{v}^s = (v_1^s, \dots, v_N^s)^T$  definiert. Jedes Partikel  $s$  merkt sich seine bisher beste Position  $\tilde{\mathbf{p}}^s$  und welches Partikel die global beste Position  $\tilde{\mathbf{p}}$  im Schwarm besitzt<sup>63</sup>. Ausgehend von diesen Informationen und seiner bisherigen Position  $\mathbf{p}^s$  und Geschwindigkeit  $\mathbf{v}^s$ , bewegt sich ein Partikel  $s \in \mathcal{S}$  in die Richtung, in der die Erfolgsaussichten hoch sind. Durch dieses simulierte soziale Verhalten des Schwarms wird das Wissen über den Suchraum verbessert, mit dem Ziel, dass sich der Schwarm dem Optimum annähert. Der Ablauf des Verfahrens ist in Algorithmus 3.7 dargestellt.

**Geschwindigkeitsupdate:** Die neue Geschwindigkeit des Partikels  $s$  ergibt sich aus seiner aktuellen Geschwindigkeit und Richtungsänderungen hin zu seiner bisher besten Position  $\tilde{\mathbf{p}}^s$  (kognitiver Anteil) und zur global besten Position  $\tilde{\mathbf{p}}$  des Schwarmes (sozialer Anteil)

$$\mathbf{v}_{neu}^s := \mathbf{v}^s + \gamma_{kog} r_{kog} (\tilde{\mathbf{p}}^s - \mathbf{p}^s) + \gamma_{soz} r_{soz} (\tilde{\mathbf{p}} - \mathbf{p}^s) \quad (3.7)$$

mit den Einflussfaktoren  $\gamma_{kog}, \gamma_{soz} > 0$  und den Zufallszahlen  $r_{kog}, r_{soz} \in [0, 1]$ . Dabei sollte die neue Geschwindigkeit

- nicht zu groß sein, da sonst die Wahrscheinlichkeit groß wird, dass ein naheliegendes Minimum übersprungen wird,
- nicht zu klein sein, da sonst der Suchraum nur sehr langsam durchschritten wird und somit auch die Konvergenz des Verfahrens langsam ist.

---

<sup>62</sup>das Optimum der OA

<sup>63</sup>bei multimodalen Aufgaben hat es sich als günstig erwiesen, anstelle der global besten Position nur die beste Position der Nachbarschaft des Partikels  $s$  zu verwenden

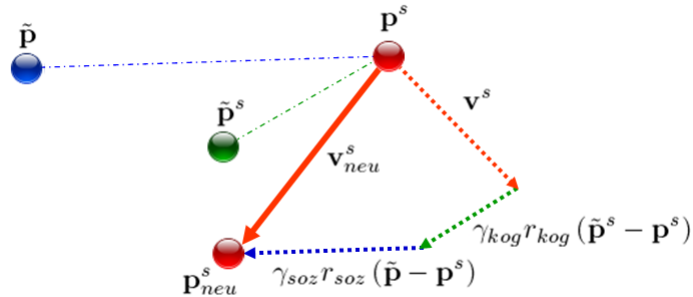


Abbildung 3.8.: Bewegung eines Partikels

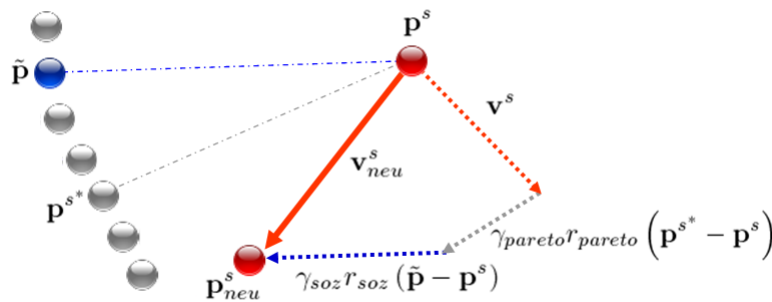


Abbildung 3.9.: schematische Bewegung eines Partikels bei MOA

Über die Einflussfaktoren  $\gamma_{kog}$ ,  $\gamma_{soz}$  kann gesteuert werden, ob eher lokal ( $\gamma_{kog} > \gamma_{soz}$ ) oder global ( $\gamma_{kog} < \gamma_{soz}$ ) gesucht werden soll. Bei zu großem sozialen Anteil  $\gamma_{soz}$  und hohen Geschwindigkeiten besteht aber auch die Gefahr, dass die Optimierung zu schnell in einem lokalen Minimum hängen bleibt, wenn das global beste Partikel sich in einem solchen befindet.

Die neue Position  $\mathbf{p}^s_{neu}$  des Partikels  $s$  resultiert aus seiner Ausgangsposition  $\mathbf{p}^s$  und seiner neuen Geschwindigkeit  $\mathbf{v}^s_{neu}$

$$\mathbf{p}^s_{neu} := \mathbf{p}^s + \mathbf{v}^s_{neu} . \quad (3.8)$$

In Abbildung 3.8 ist die Bewegung eines Partikels in einem Iterationsschritt dargestellt.

**Multikriterielle Erweiterung<sup>64</sup>:** Um die PSO für multikriterielle Optimierungsaufgaben einzusetzen, können klassische Methoden, wie sie in Abschnitt 3.2.3 beschrieben sind, eingesetzt werden. Besonders bewährt hat sich die Nutzung einer Fitnessbewertung, wie sie für Evolutionäre Verfahren (siehe Abschnitt 3.3.3.1) eingesetzt wird.

Da die Bestimmung der persönlichen besten Position  $\tilde{\mathbf{p}}^s$  eines Partikels  $s$  mittels Fitnessbewertung nicht möglich ist<sup>65</sup> (siehe auch [49]), wird die kognitive Komponente aus dem

<sup>64</sup>siehe auch [49]

<sup>65</sup>ein Partikel, dass in einer früheren Iteration den Rang 1 hatte, später zwar dichter an der PARETO-Front liegt, aber einen höheren Rang besitzt, würde zu der alten Position zurückkehren

### 3. Ausgewählte Optimierungsverfahren

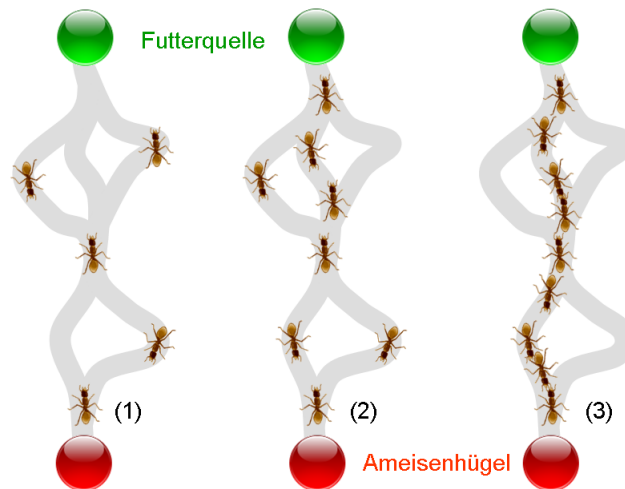


Abbildung 3.10.: Prinzip der Futtersuche einer Ameisenkolonie

- (1) ... Ameisen suchen zufällig im Zielgebiet
- (2) ... Erfolgreich zurückgekehrte Ameisen locken andere auf ihre Spur
- (3) ... die meisten Ameisen sind auf dem kürzesten Weg zwischen Hügel und Futterquelle

Geschwindigkeitsupdate 3.7 entfernt. Um zusätzlich den Sog zur PARETO-Front zu erhöhen, wird für jedes Partikel  $s \in \mathcal{S}$  ein Partikel  $s^*$  aus der PARETO-Menge zufällig ausgewählt, von dem das Partikel  $s$  ebenfalls angezogen wird (siehe Abbildung 3.9). Damit erhält man das Geschwindigkeitsupdate für multikriterielle Aufgaben

$$\mathbf{v}_{neu}^s := \mathbf{v}^s + \gamma_{soz} r_{soz} (\tilde{\mathbf{p}} - \mathbf{p}^s) + \gamma_{pareto} r_{pareto} (\mathbf{p}^{s^*} - \mathbf{p}^s) \quad (3.9)$$

mit den Einflussfaktoren  $\gamma_{soz}, \gamma_{pareto} > 0$  und den Zufallszahlen  $r_{kog}, r_{soz} \in [0, 1]$ .

#### 3.3.3.3. Ameisen Verfahren<sup>66</sup>

Ameisen sind in der Lage, ein ihnen unbekanntes Territorium sehr effizient für die Futtersuche zu erkunden. Da Ameisen so gut wie blind sind, nutzen sie hormonelle Lockstoffe, Pheromone, die jede Ameise kontinuierlich absondert, zur indirekten Kommunikation. Mit Hilfe ihrer eigenen auf dem Hinweg gelegten Pheromonspur findet sie zum einen den Rückweg zum Ameisenhügel und je näher die Ameise dem Hügel kommt, desto stärker wird die Ansammlung von Pheromon. Eine Ameise orientiert sich aber auch bei der Suche nach Futterstellen an der Pheromonkonzentration ihrer Umwelt, indem sie Wege mit hoher Konzentration bevorzugt. Ameisen, die das Gelände zufällig abgrasen, hinterlassen selbst Pheromonspuren und folgen außerdem Spuren, die andere Ameisen hinterlassen haben. Hat eine Ameise etwas von einem gefundenen Futterplatz zurückgebracht, so finden durch die hinterlassene Spur auch andere Ameisen die

<sup>66</sup>siehe [46, 57]

**Algorithmus 3.8** Ameisenalgorithmus**Eingabe** : OA

Initialisierung

```

while Abbruchbedingungen nicht erfüllt do
  | forall Ameisen do
  |   | suche zulässigen Pfad zum Ziel
  |   end
  |   Pheromon auf allen Teilpfaden verdunsten
  |   Pheromon-Update für alle neue gefundenen Pfade
end

```

Futterquelle und tragen ihren Teil zum Ameisenhügel zurück. Je mehr Ameisen dies tun, desto stärker werden die Pheromon-Spuren zur Nahrung ausgeprägt, bis schließlich immer mehr Ameisen der Route folgen. Da das Pheromon mit der Zeit auch wieder verfliegt, sind kaum benutzte oder lange Strecken weniger stark markiert als kurze und häufig benutzten Pfade, so dass künftig auch weniger Ameisen diesen Spuren folgen werden (siehe auch Abbildung 3.10).

Die Adaption des sozialen Verhaltens der Ameisen in einem Algorithmus zur Lösung kombinatorischer Probleme wurde 1991 erstmals von MARCO DORIGO<sup>67</sup> unter dem Begriff *Ant Systems* vorgestellt und in den folgenden Jahren weiterentwickelt (z.B. Ameisen-Kolonie-Verfahren, Rekrutierende Ameisen Algorithmen). Ameisenalgorithmen können gut zur Lösung kombinatorischer Probleme, die sich leicht in bewertbare Teilschritte zerlegen lassen, eingesetzt werden. Der prinzipielle Ablauf ist in Algorithmus 3.8 dargestellt.

Am Beispiel des Rundreiseproblems<sup>68</sup> lässt sich der Algorithmus gut darstellen. Die Teilschritte sind hierbei die Übergänge von einer Stadt zu einer anderen. Eine Ameise<sup>69</sup> wählt sich zufällig die nächste Stadt, wobei die Verbindungen mit hoher Pheromonkonzentration mit höherer Wahrscheinlichkeit gewählt werden als Strecken mit niedriger Konzentration. Hat die Ameise das Ziel erreicht, wird die Pheromonkonzentration aktualisiert. Dazu wird auf allen Teilstrecken das Niveau abgesenkt (Verdunstung) und anschließend der gewählte Pfad verstärkt.

Auch wenn die Vorgehensweise an die Beobachtungen in der Natur angelehnt sind, werden bewusst Änderungen vorgenommen (anders als bei vielen Evolutionären Verfahren oder Partikelschwarmverfahren). So hinterlassen die Ameisen in der Natur beispielsweise das Pheromon kontinuierlich auf den Strecken, die sie entlanglaufen, während es sich als effektiver erwiesen hat, im Algorithmus das Pheromon-Update erst dann durchzuführen, wenn ein zulässiger Pfad gefunden wurde. Weiterhin müssen die künstlichen Ameisen meist noch zusätzliche problemspezifische Informationen speichern und einbeziehen können. Dies kann am Beispiel des

<sup>67</sup> siehe auch [16]

<sup>68</sup> Ein Reisender möchte eine Anzahl von Städten auf dem kürzesten Weg nacheinander bereisen, ohne eine Stadt zweimal zu besuchen.

<sup>69</sup> manchmal auch als Agent bezeichnet

### 3. Ausgewählte Optimierungsverfahren

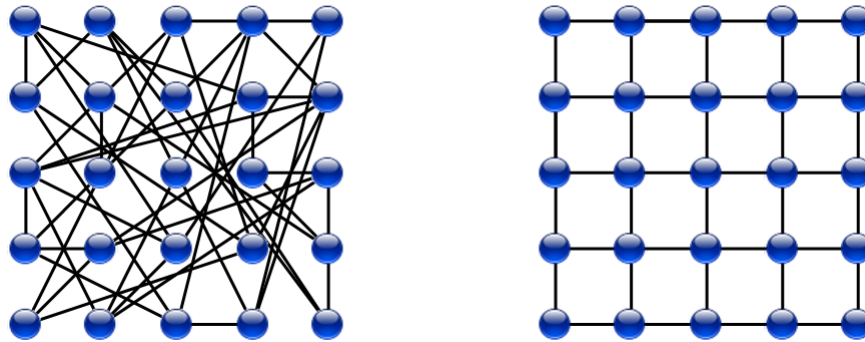


Abbildung 3.11.: Kristallgitter (verspanntes und regelmäßiges)

Rundreiseproblems eine Tabuliste mit den Orten sein, die schon besucht wurden, damit die Bedingung erfüllt wird, keine Stadt mehr als einmal zu besuchen.

Ameisenalgorithmen eignen sich gut zur Lösung kombinatorischer Problemstellungen. Damit ist eine direkte Übertragung auf komplexe technische Problemstellungen, wie sie in Abschnitt 2.2 beschrieben sind, nicht ohne weiteres möglich. Ein reellwertiger Ansatz könnte beispielsweise so aussehen, dass anstelle einzelner Teilstrecken ein geeignetes Wahrscheinlichkeitsmodell eingesetzt wird, welches die Pfade mit starken Pheromonspuren simuliert und die Parameter der zugrunde liegenden Verteilungsfunktion nur diskret betrachtet (ähnlich wie in Probabilistischen Genetischen Algorithmen, siehe [3, 58]). Auch die Erweiterung auf multikriterielle Aufgaben ist nicht ohne weiteres wie bei Evolutionären Verfahren oder Partikelschwarm-Algorithmus möglich.

#### 3.3.3.4. Simulated Annealing

Die Methode des Simulierten Abkühlens<sup>70</sup> bildet den physikalischen Vorgang des langsamen Abkühlens und der Erstarrung einer Schmelze nach. Dies wird zum Beispiel praktiziert, um möglichst homogene Kristalle aus einer Schmelze zu züchten oder Teile aus Metall oder Glas möglichst spannungsfrei zu bekommen. Dazu darf das Material nur langsam abkühlen, damit sich die Moleküle in einer gleichmäßigen, regelmäßigen Struktur anordnen können. Zu schnelle Abkühlung kann dazu führen, dass es zu Unregelmäßigkeiten in der Struktur und somit zu Verspannungen kommt. Das Simulated Annealing beschreibt den gesteuerten Erstarrungsprozess in einem Kristallgitter mit dem Ziel, einen Zustand minimaler freier Gitterenergie im Festkörper zu erreichen<sup>71</sup> (siehe Abbildung 3.11). Der physikalisch-stochastische Zusammenhang zwischen dem Energiezustand  $E(x)$  eines Körpers und seiner Temperatur  $T$  wird durch die

<sup>70</sup>engl. Simulated Annealing, 1983 von KIRKPATRICK vorgestellt, siehe auch [27]

<sup>71</sup>siehe auch [48]



**Algorithmus 3.9** Simulated Annealing**Eingabe** : OA

Initialisierung

```

while Abbruchbedingungen nicht erfüllt do
  erzeuge neuen Kandidaten  $\tilde{x}$ 
  if  $f(\mathbf{x}) > f(\tilde{x})$  then
    | akzeptiere  $\tilde{x}$ 
  else
    | akzeptiere  $\tilde{x}$  mit Wahrscheinlichkeit  $p = \exp(-\beta(f(\tilde{x}) - f(\mathbf{x})))$ 
  end
  verringere Temperatur  $T$ 
end

```

Dichtefunktion der BOLTZMANN-Verteilung

$$P_{BM}(\mathbf{x}) = c e^{\frac{-E(\mathbf{x})}{kT}}$$

beschrieben mit dem Normalisierungsfaktor  $c$  und der BOLTZMANN-Konstante  $k$ .

Dieser Vorgang wurde erfolgreich auf ein Optimierungsverfahren übertragen. Während der Optimierung wird nicht nur nach Verbesserungen gesucht, sondern es wird mit einer entsprechenden Wahrscheinlichkeit auch eine Verschlechterung der Zielfunktionswerte zugelassen. Damit wird es dem Algorithmus möglich, lokale Minima auch wieder zu verlassen und das globale Minimum zu finden. Analog dem langsamen Abkühlungsprozess einer Schmelze wird die Wahrscheinlichkeit, dass eine bestimmte Verschlechterung zugelassen wird, im Laufe der Optimierung verringert. Während so zu Beginn, also bei hoher Temperatur, noch relativ starke Verschlechterungen möglich sind, akzeptiert der Algorithmus mit sinkender Temperatur  $T$  nur noch kleine Verschlechterungen. Zu Beachten ist dabei, dass zu schnelles Abkühlen dazu führen kann, dass das Verfahren in einem lokalen Minimum hängen bleibt. Sehr langsames Abkühlen führt wiederum zu einer langsamen Konvergenz.

Der *Metropolis-Algorithmus*<sup>72</sup> bildet die Grundlage für das Simulated Annealing. Ausgehend von einer Startlösung  $\mathbf{x}$  wird eine neue Lösung  $\tilde{x}$  in der Nachbarschaft von  $\mathbf{x}$  gesucht und bei Verschlechterung mit der Wahrscheinlichkeit  $p = \exp(-\beta(f(\tilde{x}) - f(\mathbf{x})))$  mit dem Parameter  $\beta = \frac{1}{T}$  akzeptiert<sup>73</sup>. Das Verfahren ist in Algorithmus 3.9 dargestellt.

Der Bestimmung von  $\tilde{x}$  in Abhängigkeit der vorherigen Lösung  $\mathbf{x}$  und in der Nachbarschaft von  $\mathbf{x}$  erfolgt dabei so, dass der Suchraum strukturiert durchsucht wird und der Metropolis-Algorithmus eine Folge generiert, deren Elemente  $\mathbf{x}$  asymptotisch die BOLTZMANN-Verteilung mit dem Parameter  $\beta$  annähern.

Um MOA mittels Simulated Annealing zu lösen, können klassische Methoden (siehe Ab-

<sup>72</sup>1953 vorgestellt von *Nicholas Metropolis* in [39]

<sup>73</sup>bei Verbesserung wird  $\tilde{x}$  immer akzeptiert

### 3. Ausgewählte Optimierungsverfahren

---

**Algorithmus 3.10** Adaptives Ersatzmodell-Verfahren

---

**Eingabe** : OA

Startpunkte mittels LHS erzeugen

Modell erstellen

**while** *Abbruchbedingungen nicht erfüllt* **do**

    Bestimme und berechne neue erfolgversprechende Punkte

    Modelltraining (adaptive Verbesserung des Modells)

**end**

---

schnitt 3.2.3) eingesetzt werden. Eine Weiterentwicklung des SA ist die Cross-Entropy-Methode, bei der im Ablauf der Optimierung adaptiv auch die Suchverteilung angepasst wird (siehe [15, 48]).

#### 3.3.4. Adaptive Ersatzmodell-Verfahren

Bei *Ersatzmodell-Verfahren* wird ein geeignetes Approximationsmodell des Zielfunktionsraumes genutzt, um eine Optimierungsaufgabe zu lösen. Wird das Modell dabei im Laufe der Optimierung angepasst und schrittweise verbessert, spricht man von *adaptiven Ersatzmodell-Verfahren*. Da für die Initialisierung meist zufällige Werte für die Variablen genutzt werden (zum Beispiel Latin-Hypercube-Sampling), die verwendeten Modelle auf geschätzten Verteilungsfunktionen basieren und man meist stochastische Verfahren<sup>74</sup> als interne Verfahren zur Optimierung der Modellparameter einsetzt, werden auch diese Verfahren zu den stochastischen Verfahren gezählt.

Der Aufwand, der für das Modelltraining, Suche auf dem Modell und Bestimmung von neuen, vielversprechenden Punkten aufgebracht werden muss, ist im Vergleich zu anderen Verfahren<sup>75</sup> wesentlich höher. Andererseits sind diese Verfahren sehr effektiv, so dass schon mit sehr wenigen Funktionsauswertungen gute Ergebnisse erzielt werden können. Das ist insbesondere dann von Vorteil, wenn aufgrund sehr aufwändiger Zielfunktionswertberechnungen<sup>76</sup> nur wenige Varianten berechnet werden können, da der zusätzliche Verfahrensaufwand dann nur gering ins Gewicht fällt und die hohe Effektivität dieser Verfahren ausgenutzt werden kann.

Der grundlegende Ablauf einer Optimierung ist in Algorithmus 3.10 dargestellt. Nach der Initialisierung mit zufälligen Punkten<sup>77</sup> wird das Ersatzmodell erzeugt. Gut geeignet dafür ist ein *Kriging-Modell* (siehe [28, 58]). Dieses Modell wird mit den vorhandenen Informationen trainiert.

---

<sup>74</sup>häufig werden Monte-Carlo-Verfahren oder Evolutionäre Verfahren verwendet

<sup>75</sup>wie zum Beispiel naturinspirierte Verfahren

<sup>76</sup>zum Beispiel eine numerische Simulation

<sup>77</sup>zum Beispiel mittels Latin-Hypercube-Sampling

### 3.4. Bewertung der Verfahren bezüglich der Herausforderungen bei der Optimierung technischer Probleme

Aus den Schätzungen,

1. in welchen Gebieten das Modell eine hohe Unsicherheit besitzt und
2. an welchen Stellen es ein Optimum vorhersagt,

werden die vielversprechenden Punkte ermittelt, die das Modell verbessern und an das Optimum heranführen sollen. Nachdem diese Varianten berechnet sind, wird das Modell erneut trainiert und der nächste Iterationsschritt der Optimierungsschleife durchlaufen. Auf diese Weise wird das Modell im Verlauf der Optimierung adaptiv verbessert und das globale Optimum approximiert. Analog sind diese Verfahren auch für multikriterielle Aufgabenstellungen erweiterbar (siehe [58]).

### 3.4. Bewertung der Verfahren bezüglich der Herausforderungen bei der Optimierung technischer Probleme

Wesentliche Eigenschaften von OA zur Optimierung komplexer technischer Probleme wurden in Abschnitt 2.2 beschrieben. In Tabelle 3.1 werden einige wichtige dieser Eigenschaften für die beschriebenen Verfahren gegenübergestellt. Zusätzlich wurden noch die Kriterien Umgang mit diskreten und reellen Variablen hinzugefügt. Dabei steht „X“ für „es trifft zu“ und „(X)“ für „es trifft meistens oder bedingt zu“.

Der Tabelle ist zu entnehmen, dass es kein allgemein bestes Verfahren zur Lösung aller Optimierungsaufgaben gibt. Allerdings zeichnen sich bestimmte stochastische, naturinspirierte Verfahren durch ihre Einsatzfähigkeit für ein breites Spektrum von OA aus, erfordern aber einen relativ hohen Berechnungsaufwand<sup>78</sup>.

So sind beispielsweise Evolutionäre Verfahren oder Partikelschwarm-Verfahren gut geeignet, den Suchraum global zu untersuchen, benötigen aber meist viele Funktionsauswertungen, um zu einem Minimum zu konvergieren. Gradientenverfahren oder das NELDER-MEAD-Simplexverfahren hingegen finden ein Minimum mit relativ wenigen Funktionsauswertungen, enden aber im nächstbesten lokalen Minimum, so dass sie bei multimodalen Zielfunktionen das globale Minimum nicht finden, wenn die Startwerte nicht schon in dessen Nachbarschaft liegen.

Adaptive Ersatzmodell-Verfahren sind in der Lage, mit wenigen Funktionsauswertungen ein Optimum zu finden, benötigen aber zum Teil sehr hohen Aufwand, um das Modell zu trainieren und neue Vorhersagen zu treffen. Benötigen die eigentlichen Funktionsauswertungen viel Zeit, fällt das meist nicht ins Gewicht<sup>79</sup>, sind die Zielfunktionen aber mit wenig Aufwand berechenbar, sind diese Verfahren nicht mehr so effektiv, da möglicherweise ein Evolutionäres

---

<sup>78</sup>viele Funktionsauswertungen

<sup>79</sup>dauert z.B. eine Funktionsauswertung 30 Minuten, ist es unproblematisch, wenn der Optimierungsalgorithmus 30 Sekunden pro Schritt zusätzlich benötigt

### 3. Ausgewählte Optimierungsverfahren

Verfahren:	benötigt Gradienten	Umgang mit unlösbaren Varianten	Umgang mit Restriktionen	Diskrete Variablen	Reelle Variablen	viele Funktionsauswertungen nötig	hoher Verfahrensaufwand	lokale Suche	globale Suche	multikriteriell	parallelisierbar
Gradientenverfahren	X		X		X			X			
NELDER-MEAD			X	(X)	X	(X)		X			
DiRect			X	(X)	X	(X)			X		
Monte-Carlo-Methoden		X	X	X	X	X		(X)	X	X	X
Evolutionäre Verfahren		X	X	X	X	(X)		(X)	X	X	X
Partikelschwarmverfahren		X	X	X	X	(X)		(X)	X	X	X
Ameisenkolonieverfahren		X	X	X		X			X		X
Simulated Annealing		X	X	X	X	X		(X)	X		X
Adaptive Ersatzmodellverfahren		X	X	X	X		X	(X)	X	X	(X)

Tabelle 3.1.: Bewertung verschiedener Optimierungsverfahren

Verfahren in der gleichen Zeit und mit einer größeren Anzahl von Funktionsauswertungen besser konvergieren würde.

Stehen Gradienten zur Verfügung, konvergieren gradientenorientierte Verfahren in der Regel deutlich schneller als gradientenfreie Verfahren, müssen die Gradienten aber numerisch bestimmt werden, können Verfahren, die diese Information nicht benötigen, besser geeignet sein.

In vielen Fällen ist es auch von Vorteil, die Optimierung in mehrere Teilschritte zu zerlegen und diese mit unterschiedlichen Verfahren zu lösen. So kann eine erste Optimierung mit einem stochastischen Verfahren den Bereich des Optimums bzw. eine Approximation der PARETO-Front auffinden. In einer zweiten Optimierung, bei der die Ergebnisse der vorangegangenen Optimierung als Startwerte verwendet werden, wird dann mit einem geeigneten Verfahren eine Nachiteration durchgeführt, um näher an das tatsächliche Optimum heranzukommen oder die Approximation der PARETO-Front zu verbessern. Im Anschluss daran ist es möglich, mit den bekannten Informationen ein Ersatzmodell zu erzeugen und mit diesem eine weitere Optimierung durchzuführen.

# 4

## Kapitel 4.

---

# Hybride Strategien

Die Kombination verschiedener Verfahren ist einer der erfolgversprechendsten Trends in der Optimierung<sup>1</sup>. Dabei spielt insbesondere die Hybridisierung naturinspirierter, heuristischer Verfahren eine wichtige Rolle. Mit hybriden Optimierungsstrategien wird beispielsweise das Ziel verfolgt, die guten Eigenschaften verschiedener Verfahren zu vereinen, vorhandene Informationen<sup>2</sup> besser auszunutzen oder Verfahrensparameter der Optimierungsverfahren adaptiv im Verlauf einer Optimierung geeignet anzupassen.

## 4.1. Überblick

Hybride Optimierungsstrategien können hinsichtlich ihrer Methodik in vier Gruppen<sup>3</sup> unterteilt werden, wobei die Grenzen in der Praxis aber oft nicht klar gezogen werden können:

**Kombinierte Verfahren:** Bei der Kombination verschiedener Verfahren werden diese in der Regel sequentiell nacheinander ausgeführt. Im Zentrum einer solchen Methode steht die Entscheidung, zu welchem Zeitpunkt der Optimierung welches der Verfahren besser geeignet ist. Meist wird diese Entscheidung anhand statistischer und heuristischer Kennwerte getroffen. Zu diesen Strategien gehört die Methode „*Globale Suche - Lokale Verfeinerung*“, die in Abschnitt 4.2 vorgestellt wird. Ein anderer Ansatz ist die Kombination von Entscheidungsbaumverfahren<sup>4</sup> mit heuristischen Verfahren, bei denen beispielsweise ein nicht vollständiger Suchbaum zur Steuerung eines evolutionären Verfahrens verwendet wird<sup>5</sup>.

**Heuristische Strategien:** Ähnlich wie bei den naturinspirierten Optimierungsverfahren (siehe Abschnitt 3.3.3) können Beobachtungen aus der Natur, soziale Abläufe o.ä. auf Algorithmen

---

<sup>1</sup>siehe [7]

<sup>2</sup>z.B. in Form von Ersatzmodellen oder speziellen Heuristiken

<sup>3</sup>in der Literatur gibt es auch verschiedene andere Unterteilungen, die jeweils davon abhängig sind, was hervorgehoben werden soll

<sup>4</sup>z.B. Branch-And-Bound Verfahren

<sup>5</sup>siehe auch [7, 41]

#### 4. Hybride Strategien

übertragen werden, um damit den Verlauf einer Optimierung zu steuern. Die Vorgänge des Simulated Annealing, das in Abschnitt 3.3.3.4 als Optimierungsverfahren beschrieben ist, können beispielsweise auch zur adaptiven Steuerung von Verfahrensparametern im Verlauf der Optimierung genutzt werden<sup>6</sup>. Eine andere, in der Literatur beschriebene, heuristische Strategie sind Inselmodelle für Evolutionäre Verfahren. Dabei wird im frühen Stadium der Optimierung der Suchraum in meist disjunkte Teilbereiche zerlegt (Inseln), die dann separat untersucht und erst später wieder zusammengeführt werden.

**Übergeordnete Methoden:** Durch die Verbindung von Optimierungsverfahren mit anderen Methoden, zum Beispiel statistischen Analysen, können neue Verfahren mit speziellen Eigenschaften konstruiert werden. Ein Beispiel dafür ist die *Robuste Optimierung*, bei der das Ziel verfolgt wird, möglichst robuste Lösungen<sup>7</sup> zu finden, so dass das gefundene robuste Minimum nicht unbedingt auch das globale Minimum ist, sondern auch ein lokales Minimum sein kann, das aber hinsichtlich seiner Robustheit besser ist. Ein weiteres Beispiel ist die *Ersatzmodell-suche* (Abschnitt 4.3), bei der die vorhandenen Informationen dazu genutzt werden, um ein Approximationsmodell zu konstruieren und dieses Modell, bei dem die Funktionsauswertung nur noch wenig Aufwand erfordert, zu optimieren. Ein ähnlicher Ansatz wird auch bei der *Problem-Relaxation* verfolgt. Dabei wird ein vereinfachtes Modell, das leichter zu lösen ist, optimiert und die Lösung dieses Problems auf das Ausgangsproblem übertragen (beispielsweise als Startlösung für eine weitere Optimierung)<sup>8</sup>.

**Adaptive Verfahrenssteuerung:** In den verschiedenen Phasen einer Optimierung ändern sich auch die Anforderungen an das Optimierungsverfahren. So ist es zu Beginn der Optimierung wichtig, in der Breite zu suchen, um schnell die Gebiete zu identifizieren, in denen potentielle Optima zu finden sind, während es im späteren Verlauf wichtiger ist, die gefundenen Gebiete genau zu durchsuchen<sup>9</sup>. Durch adaptive Anpassung der Parameter des Optimierungsverfahren können die Eigenschaften des Verfahrens passend dazu eingestellt werden. Beispiele dafür sind die Steuerung der Populationsgröße und der Kreuzungs- sowie Mutationsraten bei Evolutionären Verfahren<sup>10</sup> oder die Abstimmung der Einflussfaktoren bei der Partikelschwarmoptimierung<sup>11</sup>.

Hybride Verfahren können auf verschiedene Weisen konstruiert werden, die sich darin unterscheiden, wie die Methoden zusammenwirken<sup>12</sup>. Die nachfolgend beschriebenen Implementierungsformen können für alle Formen der Hybridisierung umgesetzt werden.

---

<sup>6</sup>siehe Adaptive Verfahrenssteuerung

<sup>7</sup>d.h. Lösungen, deren Funktionswerte sich bei kleinen Änderungen der Variablen auch nur leicht ändern

<sup>8</sup>siehe [7, 5]

<sup>9</sup>Tiefensuche

<sup>10</sup>siehe Abschnitt 3.3.3.1

<sup>11</sup>siehe Abschnitt 3.3.3.2

<sup>12</sup>siehe z.B. [63]

**Pipeline Hybrid Verfahren:** Diese Art der Hybridisierung ist die am häufigsten eingesetzte Methode. Dabei werden die einzelnen Optimierungsverfahren oder Methoden nacheinander oder abwechselnd ausgeführt, wobei die Ergebnisse des einen Verfahrens als Startwerte des darauf folgenden verwendet werden. Die in Abschnitt 4.2 vorgestellte Methode „Globale Suche - Lokale Verfeinerung“ ist ein typischer Vertreter eines Pipeline Hybrid Verfahrens.

Eine Untergruppe dieser Methodik sind die *Hierarchischen Hybrid Verfahren*, bei denen ein Optimierungsverfahren eingesetzt wird, um beispielsweise die Steuer- oder Modellparameter eines anderen Verfahrens zu bestimmen. In diesem Sinn können auch *adaptive Ersatzmodellverfahren* (siehe Abschnitt 3.3.4) als hybride Verfahren betrachtet werden, bei denen ein Genetischer Algorithmus verwendet wird, um die Parameter des Ersatzmodells zu ermitteln.

**Asynchrone Hybrid Verfahren:** Bei diesen Methoden arbeiten zwei oder mehrere unterschiedliche Verfahren parallel, nutzen aber einen gemeinsamen Pool von Varianten<sup>13</sup>. Während die einzelnen Optimierungen unabhängig voneinander laufen, wird in regelmäßigen Abständen und dann, wenn bei einem Verfahren kein signifikanter Fortschritt mehr erzielt wird, der Austausch von Informationen über diesen Pool durchgeführt, indem gute Varianten eingebracht werden oder Varianten, die ein anderes Verfahren geliefert hat, übernommen werden.

**Hybride Operatoren:** Naturinspirierte Verfahren wie Evolutionäre Verfahren verwenden Variationsoperatoren (Kreuzung, Mutation), mit denen aus den Elternindividuen die Nachkommenindividuen erzeugt werden. Lokale Optimierungsverfahren können als zusätzliche Operatoren implementiert werden, die ausgehend von einem oder mehreren Elternpaaren optimierte Kinderindividuen ermitteln. Ähnliche Operatoren lassen sich auch für Partikelschwarmverfahren implementieren.

Auch bei diesen Ansätzen gibt es nicht „eine beste Methode“. Vielmehr ist die Wahl von dem zu lösenden Optimierungsproblem und weiteren Randbedingungen<sup>14</sup> abhängig.

## 4.2. Globale Suche - Lokale Verfeinerung

Ein Optimierungsverfahren, das gut geeignet ist, den Zielraum global zu durchsuchen und erfolgsversprechende Bereiche zu finden, ist oft nicht so gut geeignet<sup>15</sup>, das Optimum oder die PARETO-Menge genau zu bestimmen, während ein anderes Verfahren zwar schnell konvergiert, aber nur lokal arbeitet. Besonders in praktischen Anwendungen, wenn die Berechnung der Zielfunktion(en) teuer ist<sup>16</sup>, ist es von Bedeutung, mit wenigen Funktionsauswertungen dem Optimum bzw. der PARETO-Front möglichst nahe zu kommen.

<sup>13</sup>bei Evolutionären Verfahren bezeichnet man diesen Pool auch als *Shared Population*

<sup>14</sup>z.B. vorhandene Architekturen, Parallelisierung, Aufwand für die Funktionswertberechnung, usw.

<sup>15</sup>weil es dafür z.B. sehr viele Iterationen und damit Funktionsauswertungen benötigen würde

<sup>16</sup>d.h. in der Regel zeitaufwändig oder durch andere knappe Ressourcen begrenzt

#### 4. Hybride Strategien

---

**Algorithmus 4.1** Hybride Strategie: Globale Suche - Lokale Verfeinerung

---

**Eingabe** : OA

```
while Abbruchbedingungen nicht erfüllt do
|   Führe einen globalen Suchschritt aus
|   if Kriterien für lokale Suche erfüllt then
|   |   Optimierte mit lokalem Optimierer
|   |   Einfügen der Lösungen in Lösungsmenge der Globalen Suche
|
end
```

---

Mit einem global arbeitenden Verfahren, das den gesamten Suchraum durchforscht und die Gebiete identifiziert, in denen sich potentielle Minima oder PARETO-optimale Punkte befinden, ist es aber mit wenigen Funktionsauswertungen in der Regel nicht möglich, die lokalen Optima (und damit auch die globalen) ausreichend genau zu bestimmen. Lokal arbeitende Verfahren, die lokale Minima in konvexen Gebieten meist sehr effektiv auffinden können, benötigen hingegen gute Startwerte, die sich schon in der Nähe des zu findenden Optimums befinden.

Das Idee<sup>17</sup> der hybriden Methode „Globale Suche - Lokale Verfeinerung“ besteht darin, mit einem global arbeitenden Verfahren den gesamten Suchraum zu erforschen und die Gebiete mit potentiellen Optima<sup>18</sup> zu identifizieren. Sind diese ausreichend genau ermittelt, wird zu einem effektiven lokalen Verfahren gewechselt. Der schematische Ablauf ist in Algorithmus 4.1 dargestellt.

Von ausschlaggebender Bedeutung sind dabei die Entscheidungskriterien, wann zwischen den unterschiedlichen Optimierungsalgorithmen umgeschaltet wird. Bei zu spätem Umschalten auf lokale Verfeinerung wird in der globalen Suche unnötiger Aufwand generiert, wodurch die Effektivität der Methode sinkt. Wird zu früh gewechselt, sind möglicherweise die relevanten Gebiete noch gar nicht gefunden, so dass sich das globale Optimum nicht in einem der potentiell optimalen Gebiete befindet und somit bei der lokalen Suche auch nicht erreicht werden kann.

#### **Globale Suche**

Das Ziel der globalen Suche besteht darin, die Bereiche zu finden, in denen potentielle Optima erwartet werden. Populationsbasierte naturinspirierte Verfahren wie Evolutionäre Verfahren und Partikelschwarmverfahren, die in Abschnitt 3.3.3 beschrieben sind, eignen sich gut als globale Suchverfahren (siehe Tabelle 3.1 auf Seite 44). Der Vorteil dieser Verfahren besteht darin, dass mittels ihrer Heuristiken der Suchraum an vielen Stellen gleichzeitig untersucht werden kann. So ist es gut möglich, die Gebiete zu identifizieren, in denen potentielle Optima zu vermuten sind und somit in die Nähe der erwarteten Extrema zu kommen, während aber das Auffinden

---

<sup>17</sup>siehe [20]

<sup>18</sup>Einzugsbereich eines lokalen Minimums: ein lokales Verfahren konvergiert zu diesem lokalen Minimum, wenn der Startpunkt innerhalb dieses Bereichs liegt.



des exakten Optimums einen hohen Aufwand<sup>19</sup> erfordern würde.

### Lokale Suche

Eine Schwäche Evolutionärer und Partikelschwarm-Verfahren besteht darin, dass sie lokale Suchinformationen nicht ausnutzen, sondern sich nur mittels ihrer Heuristik an das Optimum herantasten. Wird eine Lösung gefunden, die dicht am Optimum liegt, muss das nicht dazu führen, dass dieses in den darauffolgenden Schritten auch tatsächlich gefunden wird.

Lokal arbeitende Suchverfahren<sup>20</sup>, wie zum Beispiel gradientenbasierte Verfahren (siehe Abschnitt 3.2.1) oder das gradientenfreie NELDER-MEAD-Simplex-Verfahren (siehe Abschnitt 3.2.2.1), konvergieren hingegen schnell zu einem Optimum, wenn sich deren Startpunkt bereits in dessen Nähe befindet<sup>21</sup>.

Durch eine Kombination von verschiedenen Verfahren für die globale und lokale Suche können die jeweiligen Stärken dieser Algorithmen symbiotisch ausgenutzt werden. Die Ausführung der lokalen Suche als zusätzlicher Operator (siehe [63]) oder im Anschluss an jeden Iterationsschritt (siehe [64]) erweist sich für Problemstellungen mit aufwändigen Funktionswertberechnungen, wie sie in Abschnitt 2.2 beschrieben sind, oft als ungünstig, da hierbei die lokale Suche auch dann durchgeführt wird, wenn möglicherweise noch gar keine erfolgversprechenden Gebiete ermittelt wurden und sich die Startwerte somit noch nicht ausreichend in der Nähe der potentiellen Optima befinden. Im ungünstigen Fall ist es möglich, dass dadurch der Aufwand sogar steigt.

Die Lösung besteht darin, dass erst dann auf die lokale Suche umgeschaltet wird, wenn die globale Suche die vielversprechenden Gebiete ausreichend gut identifiziert hat und somit geeignete Startwerte für die lokale Suche vorliegen.

#### 4.2.1. Kriterien zur Umschaltung der Suchstrategie

Für die Entscheidung, wann von der globalen auf die lokale Suche umgeschaltet wird, werden die Elemente der vorhandenen Lösungsmenge und die dazugehörigen Parameterwerte untersucht und verschiedene Kenngrößen ermittelt und interpretiert, um daraus die Entscheidung abzuleiten.

**Homogenität:** Für die Bestimmung der *relativen Homogenität*<sup>22</sup> wird die letzte Population der globalen Suche<sup>23</sup> betrachtet. Die Population wird als homogen bezeichnet, wenn die Optimierungsvariablen und Zielfunktionswerte der Individuen ähnlich sind und sich in einem kleinen Gebiet des Lösungsraumes konzentrieren. Das wird schematisch in Abbildung 4.1 dargestellt.

<sup>19</sup>d.h. viele Funktionsauswertungen

<sup>20</sup>siehe Tabelle 3.1

<sup>21</sup>d.h. es existiert kein weiteres lokales Optimum zwischen ihm und dem Startpunkt

<sup>22</sup>siehe [20]

<sup>23</sup>Evolutionäres Verfahren oder Partikelschwarmverfahren

#### 4. Hybride Strategien

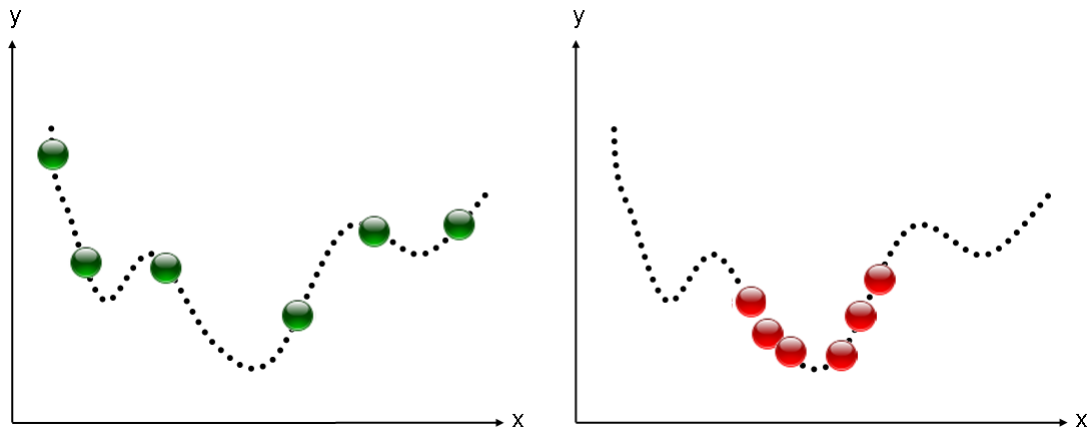


Abbildung 4.1.: Verteilung der Individuen einer Population (Beispiel)

links ... nicht homogen, die Individuen sind im ganzen Lösungsraum verteilt  
rechts ... homogen, die Individuen konzentrieren sich in einem Bereich

Während im linken Teil der Darstellung die Individuen im gesamten Lösungsraum verteilt sind, was bedeutet, dass ihre  $x$ -Werte und auch die Funktionswerte sehr verschieden sind, verdichten sie sich im rechten Bild in einem Bereich. Ihre  $x$ -Werte und auch die dazugehörigen Funktionswerte sind ähnlich.

Sind die Individuen der letzten Population  $G$  nicht homogen, deutet dies darauf hin, dass die Optimierung in vielen verschiedenen Gebieten verstreut sucht und die Lösungen sich noch nicht auf ein potentiell optimales Gebiet konzentrieren. Damit ist es sinnvoll, weiterhin mit dem globalen Optimierer zu arbeiten. Sind hingegen die Individuen recht homogen, deutet es darauf hin, dass das globale Verfahren möglicherweise ein potentiell optimales Gebiet identifiziert hat und es erfolgversprechend ist, mit einem lokalen Verfahren weiterzurechnen, um die beste Lösung in diesem Gebiet zu finden.

Zur Bewertung der Homogenität<sup>24</sup> wird der Variationskoeffizient

$$CoV = \frac{\sigma}{\mu}$$

mit dem Mittelwert

$$\mu = \frac{1}{|G|} \sum_{i=1}^{|G|} x_i$$

und der Standardabweichung

$$\sigma = \sqrt{\frac{1}{|G| - 1} \sum_{i=1}^{|G|} (x_i - \mu)^2}$$

<sup>24</sup>siehe auch [20, 21]

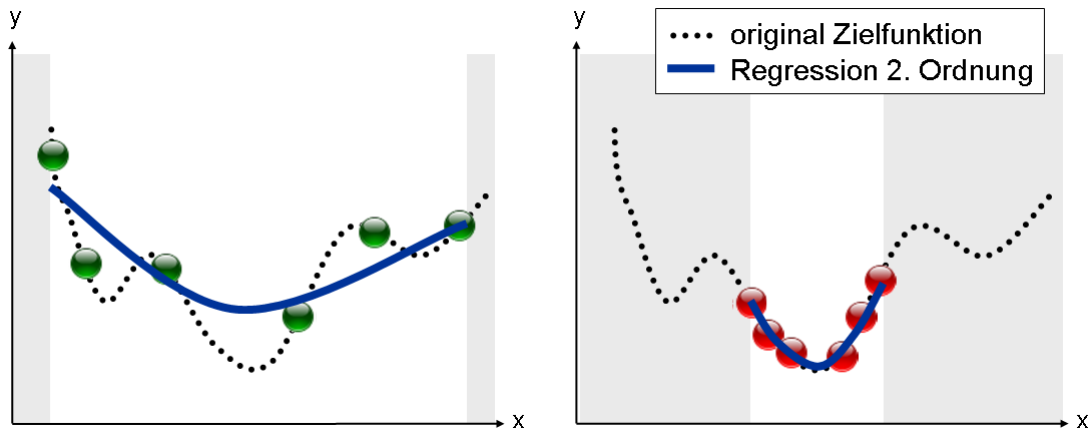


Abbildung 4.2.: Regressionsanalyse der Individuen einer Population (Beispiel)

links ... Regression fittet den betrachteten Bereich des Zielraumes schlecht  
rechts ... Regression fittet den betrachteten Bereich des Zielraumes gut

der Variablen und Zielfunktionswerte der Individuen der letzten Populationen betrachtet. Wenn der Variationskoeffizient in der aktuellen Population anwächst, weist das darauf hin, dass der Optimierer neue Bereiche untersucht und somit die globale Optimierungsstrategie besser geeignet ist. Sinkt er hingegen, zeigt dies an, dass die Homogenität der Population zunimmt und sich die Effizienz der Optimierung mit der lokalen Strategie erhöhen lässt.

**Regressionsanalyse:** Eine weitere Methode, die von KURT HACKER et.al. in [20] und [21] vorgestellt wurde und die für die Umschalt-Entscheidung genutzt werden kann, basiert auf einer Regressionsanalyse der Zielfunktionswerte (bzw. Fitnesswerte) der Individuen der letzten Population. Dazu wird ein quadratisches Polynommodell

$$\tilde{f}(\mathbf{x}) = \tilde{f}(x_1, x_2, \dots) = A_1x_1^2 + A_2x_2^2 + \dots + B_1x_1x_2 + \dots + C$$

erzeugt und dessen Parameter mittels der Methode der kleinsten Quadrate bestimmt. Zur Bewertung werden die quadratischen Abstände der gegebenen  $N^*$  Punkte von der Regressionsfunktion berechnet und aufsummiert:

$$\epsilon = \sum_{n=1}^{N^*} \left( f(\mathbf{x}^{(n)}) - \tilde{f}(\mathbf{x}^{(n)}) \right)^2 .$$

Die Bewertungs-idee ist in Abbildung 4.2 an einem Beispiel dargestellt. Sind bei einer komplizierten Zielfunktion die Individuen in einem weiten Bereich des Lösungsraumes verstreut, ist der Fehler  $\epsilon$  des Regressionsmodells groß (siehe linke Darstellung). Das globale Optimierungsverfahren sollte somit weiter verwendet werden. Sind die Individuen in einem potentiell optimalen Bereich konzentriert und damit der Fehler  $\epsilon$  des Regressionsmodells klein (siehe

#### 4. Hybride Strategien

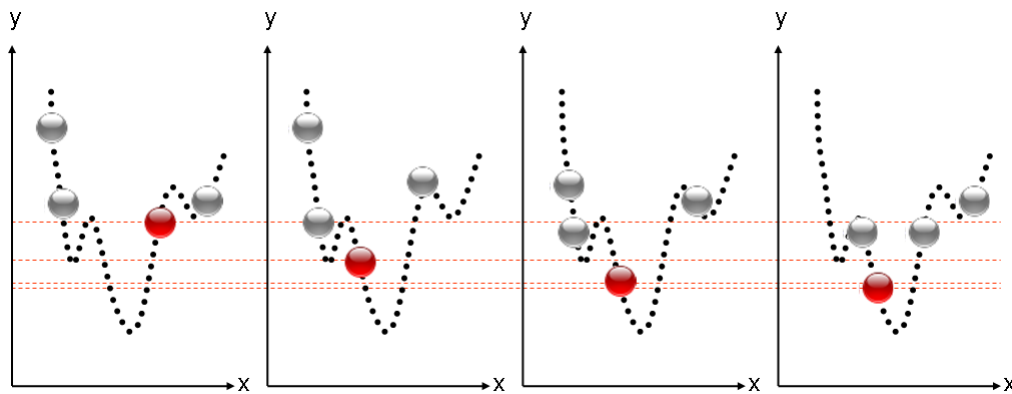


Abbildung 4.3.: Fortschritt der Optimierung (Beispiel)

rechte Darstellung), deutet dies darauf hin, dass es effektiver sein wird, den lokalen Optimierer zu verwenden, um das (lokale) Optimum innerhalb dieses Bereiches mit wenig Aufwand zu ermitteln.

**Fortschrittsanalyse:** Die Fortschrittsanalyse wird eingesetzt, um festzustellen, ob die Optimierung mit dem globalen Optimierer von Generation zu Generation Verbesserungen findet oder ob sie stagniert. Besonders dann, wenn ein potentiell optimaler Bereich recht klein ist, kann es sehr lange dauern, bis das globale Verfahren soweit konvergiert ist, dass sich die gesamte Population in diesem Bereich befindet. Wird eine Stagnation festgestellt, sollte ebenfalls auf den lokalen Optimierer umgeschaltet werden.

In Abbildung 4.3 ist dieser Zusammenhang an einem Beispiel dargestellt. Während zu Beginn der Optimierung (linke Seite) noch große Verbesserungen gefunden werden, stagniert die Suche auf der rechten Seite.

Um den Fortschritt der Optimierung zu bewerten, wird der Gradient der Verbesserung über die letzten Generationen berechnet. Geht er für mehrere aufeinanderfolgende Generationen gegen null, ist das ein Anzeichen für eine Stagnation der Optimierung. Zu beachten ist bei diesem Kriterium allerdings, dass bei ungünstiger Parametrisierung des globalen Optimierers<sup>25</sup> oder ungünstiger Zielfunktion<sup>26</sup> zwischen zwei aufeinanderfolgenden Generationen möglicherweise keine oder nur eine geringe Verbesserung erzielt wird, obwohl sich die Optimierung noch in einem Stadium befindet, wo der globale Optimierer eingesetzt werden sollte. Deshalb ist es angebracht, für die Berechnung des Gradienten der Verbesserung die aktuelle und eine ältere Generation zu verwenden.

<sup>25</sup>z.B. eine sehr kleine Populationsgröße oder sehr geringe Variationsrate

<sup>26</sup>z.B. viele „schmale“ zulässige Bereiche oder viele unlösbare Varianten (insbesondere zu Beginn einer Optimierung)

## 4.2.2. Multikriterielle Erweiterung

Die in Abschnitt 4.2.1 beschriebenen Kriterien zur Umschaltung zwischen globaler und lokaler Suche sind gut geeignet für Optimierungsaufgaben mit einer Zielfunktion. Für den Einsatz in MOA werden angepasste oder neue Kriterien benötigt. Weiterhin ist beim Einsatz der lokalen Optimierungsstrategie zu beachten, dass es sich bei den hier eingesetzten lokalen Optimierern um deterministische, einkriterielle Verfahren handelt, die nur durch den Einsatz von klassischen Methoden (siehe Abschnitt 3.2.3) auf MOA angewendet werden können. Wichtig ist dabei die Auswahl einer geeigneten Ersatzoptimierungsaufgabe.

### 4.2.2.1. Multikriterielle Umschaltkriterien

**Homogenität der Fitness:** Die Lösung einer MOA ist die PARETO-Menge, die die möglichen Kompromisse zwischen den Zielfunktionen enthält. Einige oder auch alle Individuen<sup>27</sup> einer Population des globalen Optimierers stellen stets eine Approximation der PARETO-Menge dar. Daher kann die Homogenität der Parameter und Zielfunktionen auch dann nicht höher werden, wenn die Optimierung zur PARETO-Front konvergiert. Deshalb wird für multikriterielle OA die Homogenität der Fitnesswerte untersucht. Enthält eine Population Individuen sowohl mit hoher und niedriger Fitness, deutet dies darauf hin, dass die Optimierung noch nicht konvergiert und die Population somit auch viele nicht PARETO-optimale Lösungen enthält. Dann ist es auch sinnvoll, weiterhin den globalen Optimierer zu verwenden. Haben alle Individuen eine hohe Fitness<sup>28</sup>, ist der Einsatz der lokalen Suche sinnvoll, um verschiedene Stellen der PARETO-Front näher zu untersuchen (siehe Abschnitt 4.2.2.2).

**Fortschrittsanalyse:** Der Fortschritt einer MOA kann daran gemessen werden, dass die approximierte PARETO-Menge der aktuellen Population dichter an der tatsächlichen PARETO-Menge liegt als die Approximation einer älteren Population. Da aber die exakte PARETO-Menge nicht bekannt ist, ist es im Verlauf der Optimierung auch nicht möglich, die aktuelle Näherung damit zu vergleichen.

Abhilfe schafft ein Relativvergleich der PARETO-Mengen der aktuellen und einer früheren Population. Dominiert die PARETO-Menge der aktuellen Population die der früheren Population, signalisiert dies einen Fortschritt in der Optimierung und der globale Optimierer sollte weiter eingesetzt werden. Kann keine signifikante Dominanz festgestellt werden, deutet dies darauf hin, dass die PARETO-Front erreicht ist oder die Optimierung stagniert. Dann ist wiederum der Einsatz der lokalen Suche sinnvoll, um verschiedene Stellen der Front näher zu untersuchen (siehe Abschnitt 4.2.2.2).

In Abbildung 4.4 ist dieser Zusammenhang mit einem Beispiel von zwei zu minimierenden Zielfunktionen dargestellt. In der linken Darstellung ist eine deutliche Verbesserung der PARETO-Menge der aktuellen Population gegenüber der einer früheren Population erkennbar. In der

<sup>27</sup>abhängig vom Fortschritt der Optimierung

<sup>28</sup>das entspricht gleichzeitig einer hohen Homogenität der Fitness

#### 4. Hybride Strategien

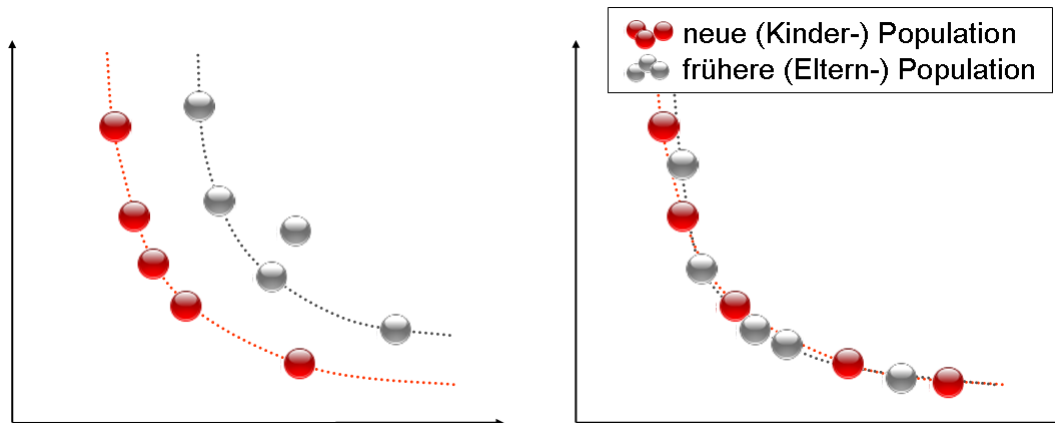


Abbildung 4.4.: Fortschritt einer multikriteriellen Optimierung (Beispiel mit 2 ZF)

links ... die aktuelle (neue) Population dominiert die Vergleichs- (alte) Population

rechts ... die aktuelle (neue) Population dominiert die Vergleichs- (alte) Population nicht

Rechten liegen die PARETO-Mengen nahezu übereinander, so dass kein Fortschritt erkennbar ist.

Für den Relativvergleich werden zuerst die PARETO-Mengen der beiden zu vergleichenden Populationen vereinigt

$$\check{M} = X_{akt} \cup X_{alt}$$

und alle dominierten Individuen aus der Vereinigung  $\check{M}$  entfernt. Anschließend wird gezählt, wie viele Individuen der resultierenden PARETO-Menge  $\check{X}$  jeweils aus den einzelnen Populationen stammen:

$$i_{akt} = \sum_{\mathbf{x} \in X_{akt}} \begin{cases} 1 & , \text{ wenn } \mathbf{x} \in \check{X} \\ 0 & , \text{ sonst} \end{cases}$$

$$i_{alt} = \sum_{\mathbf{x} \in X_{alt}} \begin{cases} 1 & , \text{ wenn } \mathbf{x} \in \check{X} \\ 0 & , \text{ sonst} \end{cases} .$$

Wenn gilt  $i_{akt} > i_{alt}$ , dominiert die aktuelle Population die ältere, es ist Fortschritt zu erkennen.

Ähnlich wie beim einkriteriellen Fortschrittskriterium ist darauf zu achten, dass bei ungünstiger Parametrisierung des Optimierungsverfahrens der Fortschritt zwischen zwei aufeinanderfolgenden Generationen kein zuverlässiges Kriterium ist und deshalb besser die aktuelle mit einer früheren Population verglichen werden sollte. Wird allerdings mit einer zu alten Population verglichen, wird eine mögliche Konvergenz oder Stagnation erst relativ spät erkannt.

#### 4.2.2.2. Auswahl der Ausgangspunkte für die lokale Suche

Ist bei einer MOA ein Kriterium zur Umschaltung auf die lokale Suche erfüllt, dann ist, anders als bei einer EOA, bei der das Optimum des globalen Optimierers als Ausgangspunkt der

lokalen Optimierung verwendet wird, die Entscheidung nicht automatisch gegeben, an welchen Stellen lokal weitergesucht wird. Vielmehr müssen sinnvolle Startpunkte und passende Ersatzzielfunktionen festgelegt werden.

Die Auswahl der Startpunkte kann erfolgen, indem

- die Individuen mit der höchsten Fitness ausgewählt oder
- die signifikanten Vertreter der PARETO-Menge mittels Clusteranalyse ermittelt werden<sup>29</sup>.

Der lokale Optimierer wird dann genutzt, um die Fitnessfunktion zu maximieren. Auf diese Weise wird die Konvergenz in Richtung PARETO-Menge vorangetrieben. Alternativ ist es auch möglich, von mehreren Startpunkten und mit verschiedenen Ersatzzielfunktionen (siehe Abschnitt 3.2.3) zu starten. Um dabei nicht nur einen speziellen Kompromiss der PARETO-Menge zu betrachten, muss die lokale Suche mehrfach von verschiedenen Startpunkten und mit verschiedenen Parametrisierungen<sup>30</sup> der Ersatzzielfunktion ausgeführt werden.

#### 4.2.2.3. Suche mit eingeschränkten Wertebereichen der Optimierungsvariablen

Eine Alternative zu diesem Vorgehen kann aber auch sein, dass anstelle des lokalen Optimierers weiterhin der globale Optimierer verwendet wird und dafür die Wertebereiche der Optimierungsvariablen so eingeschränkt werden, dass der Suchraum dadurch deutlich verkleinert wird. Dieses Vorgehen bringt aber nur dann einen Vorteil, wenn die Werte der Variablen der Individuen, die zur PARETO-Menge gehören, tatsächlich alle in einem eingeschränkten Wertebereich liegen.

## 4.3. Ersatzmodellssuche

Ein gut abgestimmtes Ersatzmodell, das die Realität mit einer ausreichenden Genauigkeit widerspiegelt und mit welchem die Funktionswerte mit wenig Aufwand berechnet werden können, kann anstelle der tatsächlichen Zielfunktionen eingesetzt werden, um aufwändige Zielfunktionsauswertungen einzusparen. Da aber auch ein gutes Ersatzmodell stets einen Modellfehler und eine Modellunsicherheit besitzt, können die Ergebnisse dieser Suche nicht direkt als Ergebnis der Optimierung verwendet werden, sondern es werden ausgewählte Vertreter mit den Originalzielfunktionen nachgerechnet und dann zur Lösungsmenge hinzugefügt.

Für *adaptive Ersatzmodellverfahren*, die in Abschnitt 3.3.4 beschrieben sind, ist das Ersatzmodell der zentrale Kern des Optimierungsverfahrens und wird in jedem Schritt adaptiv verbessert und zur Vorhersage neuer potentiell optimaler Lösungen verwendet. Die dabei häufig eingesetzten Approximationsverfahren<sup>31</sup> müssen eine hohe Güte haben und erfordern oft einen hohen Berechnungsaufwand. Dafür benötigt das Optimierungsverfahren aber meist nur relativ wenige Funktionsauswertungen, um das Optimum oder die PARETO-Menge zu finden.

---

<sup>29</sup>siehe auch [28]

<sup>30</sup>z.B. PARETO-Scan

<sup>31</sup>z.B. Kriging

## 4. Hybride Strategien

Einige der Vorteile dieser Verfahren können auch in hybriden Strategien implementiert werden, um Funktionsauswertungen einzusparen oder die Konvergenz zu verbessern.

### 4.3.1. Approximationsverfahren

Als Ersatzmodell wird eine Funktion  $\hat{f}(\mathbf{x})$  gesucht, die eine gegebene Funktion  $f(\mathbf{x})$ , von der  $n$  Wertetupel  $(\mathbf{x}^{(i)}, f(\mathbf{x}^{(i)}))$ ,  $i = 1, \dots, n$  bekannt sind, möglichst gut approximiert. Wird die Forderung gestellt, dass alle Wertetupel  $(\mathbf{x}^{(i)}, f(\mathbf{x}^{(i)}))$  exakt reproduziert werden, so dass gilt  $\hat{f}(\mathbf{x}^{(i)}) \equiv f(\mathbf{x}^{(i)})$ ,  $\forall i = 1, \dots, n$ , spricht man von Interpolation.

Eine umfangreiche Untersuchung von geeigneten Approximations- und Interpolationsverfahren für den Einsatz in technischen Aufgabenstellungen ist in [59] dargestellt.

Für die hybride Optimierungsstrategie wurde in dieser Arbeit das *Moving-Least-Squares-Verfahren* (siehe Abschnitt 4.3.1.2) ausgewählt, welches sich durch eine hohe Güte bei geringem Trainingsaufwand auszeichnet. Es stellt eine Weiterentwicklung der multiplen Regression durch Lokalisierung des Approximationsansatzes dar<sup>32</sup>.

#### 4.3.1.1. Multiple Regression

Als *Multiple Regression* wird die mehrdimensionale Approximation im quadratischen Mittel bezeichnet, bei der eine Approximationsfunktion

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^m c_i \phi_i(\mathbf{x}) \quad (4.1)$$

mit den  $m$  Basisfunktionen  $\phi_i(\mathbf{x})$ ,  $i = 1, \dots, m$  und den Koeffizienten  $c_i$ ,  $i = 1, \dots, m$  verwendet wird.

Als Basisfunktionen werden die mehrdimensionalen Tensorprodukte der eindimensionalen, normierten LEGENDRE-Polynome<sup>33</sup> eingesetzt (siehe [1, 59]).

Mit den gegebenen  $n$  Wertetupeln  $(\mathbf{x}^{(i)}, f(\mathbf{x}^{(i)}))$ ,  $i = 1, \dots, n$  und den  $m$  Basisfunktionen  $\phi_i(\mathbf{x})$ ,  $i = 1, \dots, m$  erhält man das überbestimmte Gleichungssystem

$$\Phi \mathbf{c} = \mathbf{b} \quad (4.2)$$

zur Bestimmung der Koeffizienten  $c_i$ ,  $i = 1, \dots, m$  mit

$$\Phi = \begin{pmatrix} \phi_1(\mathbf{x}^{(1)}) & \phi_2(\mathbf{x}^{(1)}) & \cdots & \phi_m(\mathbf{x}^{(1)}) \\ \phi_1(\mathbf{x}^{(2)}) & \phi_2(\mathbf{x}^{(2)}) & \cdots & \phi_m(\mathbf{x}^{(2)}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}^{(n)}) & \phi_2(\mathbf{x}^{(n)}) & \cdots & \phi_m(\mathbf{x}^{(n)}) \end{pmatrix}, \mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{pmatrix}, \mathbf{b} = \begin{pmatrix} f(\mathbf{x}^{(1)}) \\ f(\mathbf{x}^{(2)}) \\ \vdots \\ f(\mathbf{x}^{(n)}) \end{pmatrix}.$$

<sup>32</sup>siehe [30]

<sup>33</sup>siehe Anhang A



Dieses Gleichungssystem kann durch Überführung in ein GAUSSSches Normalgleichungssystem

$$\Phi^T \Phi \mathbf{c} = \Phi^T \mathbf{b} \quad (4.3)$$

oder durch Einsatz des HOUSEHOLDER-Verfahrens direkt gelöst werden<sup>34</sup>.

Ein Nachteil der Multiplen Regression besteht darin, dass die Basisfunktionen für den gesamten Definitionsbereich gelten und somit der globale quadratische Fehler der Approximation minimiert wird.

#### 4.3.1.2. Moving-Least-Squares-Verfahren<sup>35</sup>

Insbesondere dann, wenn die Funktion viele lokale Minima, Sprünge o.ä. besitzt, empfiehlt es sich, die Fehlerminimierung zu lokalisieren. Beim *Moving-Least-Squares-Verfahren (MLS)* werden die skalaren Koeffizienten  $c_i$ ,  $i = 1, \dots, m$  durch Koeffizientenfunktionen  $c_i(\mathbf{x})$ ,  $i = 1, \dots, m$  ersetzt. Aus 4.1 erhält man dann die Approximationsfunktion

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^m c_i(\mathbf{x}) \phi_i(\mathbf{x}). \quad (4.4)$$

Um eine Lokalisierung der Approximation zu erreichen, wird eine Gewichtsfunktion eingeführt, die für eine gesuchte Stelle  $\mathbf{x}^*$  die Stützstellen  $\mathbf{x}^{(j)}$ ,  $j = 1, \dots, n$  entsprechend ihres Abstandes von  $\mathbf{x}^*$  wichtet. In [61] wird dafür die GAUSSfunktion

$$w(\mathbf{x}^{(i)}) = \exp\left(-\frac{\|\mathbf{x}^* - \mathbf{x}^{(i)}\|^2}{\lambda^2}\right)$$

vorgeschlagen, wobei mit dem Parameter  $\lambda$  die Größe des Einflussbereiches gesteuert werden kann.

Damit erhält man mit

$$\phi(\mathbf{x}^*) = \begin{pmatrix} \phi_1(\mathbf{x}^*) \\ \vdots \\ \phi_m(\mathbf{x}^*) \end{pmatrix}, \quad \mathbf{W} = \begin{pmatrix} w(\mathbf{x}^{(1)}) & & & 0 \\ & w(\mathbf{x}^{(2)}) & & \\ & & \ddots & \\ 0 & & & w(\mathbf{x}^{(n)}) \end{pmatrix}$$

das Approximationsschema

$$\hat{f}(\mathbf{x}^*) = \phi(\mathbf{x}^*) (\Phi^T \mathbf{W} \Phi)^{-1} \Phi^T \mathbf{W} \mathbf{b} \quad (4.5)$$

<sup>34</sup>siehe auch [59, 17]

<sup>35</sup>siehe [59, 30]

#### 4. Hybride Strategien

zur Bestimmung des Funktionswertes der Approximationsfunktion und die Koeffizientenfunktion

$$\mathbf{c}(\mathbf{x}^*) = (\Phi^T \mathbf{W} \Phi)^{-1} \Phi^T \mathbf{W} \mathbf{b}. \quad (4.6)$$

Da  $\mathbf{W}$  und somit auch  $\mathbf{c}$  von der gesuchten Position  $\mathbf{x}^*$  abhängig sind, müssen sie für die Ermittlung jedes gesuchten Funktionswertes neu berechnet werden. Dazu wird das Gleichungssystem

$$\Phi^T \mathbf{W} \Phi \mathbf{c} = \Phi^T \mathbf{W} \mathbf{b} \quad (4.7)$$

gelöst. Der Berechnungsaufwand für einen Funktionswert ist dadurch deutlich höher als bei der Multiplen Regression.

#### 4.3.2. Optimierung mit Ersatzmodell und Nachrechnung

Als hybride Methode kann die Ersatzmodellsuche eingesetzt werden, um

- während der Optimierung (zeitweise) mit einem Ersatzmodell weiterzurechnen, wenn dieses ausreichend genau ist, um die Konvergenz zu beschleunigen und aufwändige Zielfunktionsauswertungen einzusparen, oder
- im Anschluss an eine Optimierung mit einem Ersatzmodell nachzuiteilieren, um beispielsweise die Dichte der PARETO-Menge zu erhöhen oder zu versuchen, die gefundene Lösung noch zu verbessern.

Die Ersatzmodellsuche ist dabei aber stets ein Zwischenschritt, da gefundene Lösungen mit den Originalzielfunktionen nachgerechnet werden müssen.

Der prinzipielle Ablauf dieser Methoden ist in Algorithmus 4.2 und in Algorithmus 4.3 dargestellt.

##### 4.3.2.1. Ersatzmodellsuche während der Optimierung

Ähnlich wie bei den *adaptiven Ersatzmodellverfahren* wird ein Approximationsmodell anstelle der Zielfunktionen verwendet, um das Optimum bzw. die PARETO-Menge der Optimierungsaufgabe zu bestimmen. Voraussetzung dafür ist, dass das Ersatzmodell eine ausreichende Genauigkeit besitzt.

Um zu entscheiden, ob eine Umschaltung in die *Ersatzmodellsuche* sinnvoll ist, muss ein geeignetes Bewertungskriterium definiert werden. Dazu werden die vorhandenen Lösungen in eine Trainings- und eine Testmenge geteilt und das Approximationsmodell (mit der Trainingsmenge) erzeugt. Zur Bewertung wird das Bestimmtheitsmaß<sup>36</sup>

$$R^2 = 1 - \frac{\sum_{i=1}^n (y^{(i)} - \hat{y}^{(i)})^2}{\sum_{i=1}^n (y^{(i)} - \mu)^2} \quad (4.8)$$

---

<sup>36</sup>siehe [31, 59]

**Algorithmus 4.2** Hybride Strategie: Ersatzmodellssuche innerhalb der Optimierung

---

**Eingabe** : OA

Initialisierung

**while** *Abbruchbedingungen nicht erfüllt* **do**

    Führe einen globalen Suchschritt aus

    Erzeuge Ersatzmodell (Moving-Least-Squares)

**if** *Modell ist ausreichend gut* **then**

        Optimiere mit Ersatzmodell

        Nachrechnung mit Original-Zielfunktionen

        Einfügen der Lösungen in die Lösungsmenge der Globalen Suche

**end**

---

berechnet, wobei  $\hat{y}^{(i)}$  die Schätzungen des Modells sind und  $\mu$  das arithmetische Mittel der Originalpunkte ist. Zur Berechnung von  $R^2$  werden die Testpunkte oder alle Punkte der Trainings- und Testmenge verwendet. Das Bestimmtheitsmaß ist  $R^2 = 1$  bei perfekter Übereinstimmung und  $R^2 \leq 0$ , wenn keine oder nur sehr geringe Übereinstimmung<sup>37</sup> erkannt wird.

Nach jedem Iterationsschritt der globalen Optimierung<sup>38</sup> wird aus den bekannten Informationen ein Ersatzmodell mittels der *Moving-Least-Squares-Methode* erzeugt und das Bestimmtheitsmaß berechnet. Ist dieses größer als eine vorgegebene Schranke  $\varepsilon_{R^2}$ , wird ein Schritt der *Ersatzmodellssuche* gestartet. Aus der OA in (2.1) wird durch Substitution der Zielfunktionen und Nebenbedingungen mit den Approximationsfunktionen des Modells die Ersatzaufgabe

$$(NLMOA) \begin{cases} \hat{f}_m(\mathbf{x}) \rightarrow \min & m = 1, 2, \dots, M \\ \hat{g}_j(\mathbf{x}) \geq 0 & j = 1, 2, \dots, J \\ x_i^{(U)} \leq x_i \leq x_i^{(O)} & i = 1, 2, \dots, N \\ \mathbf{x} \in \mathbb{R}^N \end{cases} \quad (4.9)$$

Die OA 4.9 wird dann mit einem globalen Optimierungsalgorithmus mit geeigneter Parametrisierung<sup>39</sup> gelöst. Die Lösung muss anschließend mit den Originalzielfunktionen nachgerechnet werden. Handelt es sich um eine MOA mit mehreren Zielen, wird eine vorgegebene Anzahl von Lösungen der PARETO-Menge ausgewählt und berechnet.

Die nachgerechneten Lösungen der Ersatzmodellssuche werden zur aktuellen Population des globalen Optimierers hinzugefügt und die Optimierung wird fortgesetzt.

#### 4. Hybride Strategien

---

**Algorithmus 4.3** Hybride Strategie: Ersatzmodellsuche als Nachiteration

---

**Eingabe** : OA

Führe Optimierung aus

Erzeuge Ersatzmodell (Moving-Least-Squares)

**if** *Modell ist ausreichend gut* **then**

    Optimiere mit Ersatzmodell

    Nachrechnung mit Original-Zielfunktionen

    Einfügen der Lösungen in die Lösungsmenge der Globalen Suche

---

#### 4.3.2.2. Ersatzmodellsuche als Nachiteration

Nach Abschluss einer Optimierung kann es sinnvoll sein, mit den vorhandenen Informationen ein geeignetes Ersatzmodell zu erzeugen. Wie bereits in Abschnitt 4.3.2.1 beschrieben, ist die substituierte OA (4.9) zu lösen. Die nachgerechneten Ergebnisse werden anschließend zur Lösungsmenge hinzugefügt.

Besonders dann, wenn nur wenige Funktionsauswertungen während der Optimierung möglich sind<sup>40</sup>, und die Optimierung nicht ausreichend konvergiert ist, kann die Nachiteration mit einem guten Ersatzmodell großes Verbesserungspotential bedeuten. Ist es nicht möglich, ein gutes Modell zu erzeugen<sup>41</sup>, muss abgeschätzt werden, inwieweit es sinnvoll ist, die Ersatzmodellsuche auszuführen und deren Ergebnisse nachzurechnen. Die Schranke<sup>42</sup>  $\varepsilon_{R^2}^{(ende)}$  wird dabei kleiner gewählt werden als  $\varepsilon_{R^2}$ .

### 4.4. Bewertung der Strategien bezüglich der Herausforderungen bei der Optimierung technischer Probleme

Die in diesem Abschnitt vorgestellten hybriden Strategien sind für den Einsatz in komplexen technischen Aufgabenstellungen gut geeignet. Wie Tabelle 4.1 zu entnehmen ist, können alle Strategien eingesetzt werden, um die Effektivität eines Optimierungsverfahrens zu verbessern. Aber es gibt auch Einschränkung, die beachtet werden müssen. Während die Methode „Globale Suche - Lokale Verfeinerung“ gut für Aufgaben mit wenigen Optimierungsvariablen eingesetzt werden kann, sinkt der Nutzen dieser Strategie mit der Anzahl der Optimierungsvariablen. Bei Aufgaben mit stark eingeschränktem Parameterraum<sup>43</sup> sind alle Hybridmethoden zwar

---

<sup>37</sup>ohne Einbeziehung der Testmenge liegt  $R^2$  im Intervall [0;1]

<sup>38</sup>abgeschlossene Berechnung einer Generation

<sup>39</sup>Die Parametrisierung des Verfahrens sollte so gewählt werden, dass eine gute Konvergenz (Optimum oder PARETO-Menge) erreicht wird.

<sup>40</sup>z.B. lange Rechenzeiten für eine Zielfunktionsberechnung und begrenzte Optimierungszeit

<sup>41</sup>das Bestimmtheitsmaß ist  $R^2 \ll 1$

<sup>42</sup>für die Entscheidung, ob die Ersatzmodellsuche als Nachiteration ausgeführt wird

<sup>43</sup>sehr enge Restriktionen

#### 4.5. Das hybride Optimierungsverfahren AHMOS

Hybrid-Strategie:	ein Ziel	mehrere Ziele	sehr viele Variablen	stark beschränkter Parameterraum	viele unlösbare Varianten	keine Gradienten verfügbar	viele lokale Extrema	sehr wenige Funktionsauswertungen möglich
Globale Suche / Lokale Verfeinerung	X	(X)		(X)		X	(X)	(X)
Ersatzmodellsuche während der Optimierung	X	X	(X)	(X)	X	X	(X)	X
Nachiteration mit Ersatzmodellsuche	X	X	X	(X)	X	X	(X)	X

Tabelle 4.1.: Bewertung verschiedener hybriden Strategien

einsetzbar, haben aber auch ihre Grenzen, besonders dann, wenn der zulässige Bereich aus mehreren, engen und nichtzusammenhängenden Gebieten besteht.

Wenn nur sehr wenige Funktionswerte berechnet werden können, ist die „Ersatzmodellsuche“ sehr effektiv einzusetzen.

Für Optimierungsaufgaben, bei denen die Funktionswerte an vielen Stellen nicht berechnet werden können (unlösbare Varianten), ist die „Globale Suche - Lokale Verfeinerung“ oft nicht oder nur schlecht einsetzbar, da der lokale Optimierer dadurch stark behindert wird. Ist die Stabilität einer Simulationssoftware für die Berechnung der Zielfunktionswerte die Ursache dafür, kann die „Ersatzmodellsuche“ die Effektivität stark erhöhen.

Bei der Konstruktion eines hybriden Optimierungsverfahrens ist auf geeignete Entscheidungskriterien für den Einsatz der jeweiligen Methoden zu achten.

### 4.5. Das hybride Optimierungsverfahren AHMOS<sup>44</sup>

Im Optimierungsverfahren<sup>45</sup> AHMOS (*Adaptive Hybride Multi-criterion Optimization Strategy*) sind die in diesem Kapitel untersuchten hybriden Strategien implementiert. Als Optimierungsverfahren kommen im Rahmen dieser Arbeit der Genetische Algorithmus *NSGA-II* und das *NELDER-MEAD-Simplex-Verfahren* zum Einsatz.

<sup>44</sup> Adaptive Hybride Multi-criterion Optimization Strategy

<sup>45</sup> dieses Verfahren wurde im Rahmen dieser Arbeit entwickelt

## 4. Hybride Strategien

Wie der Tabelle 3.1 zu entnehmen ist, können auch andere Evolutionäre Verfahren (z.B. Probabilistische Genetische Algorithmen<sup>46</sup>) oder Partikelschwarm Verfahren<sup>47</sup> als globale Optimierer eingesetzt werden. Für einkriterielle Aufgaben ist auch das Verfahren DiRect<sup>48</sup> als globaler Optimierer gut geeignet. Stehen Gradienten zur Verfügung, bietet es sich außerdem an, anstelle des NELDER-MEAD-Simplex-Verfahrens ein gradientenbasiertes Verfahren<sup>49</sup> als lokalen Optimierer einzusetzen.

Adaptive Ersatzmodellverfahren sind prinzipiell auch als globale Optimierer einsetzbar. Aufgrund ihrer speziellen Arbeitsweise<sup>50</sup> (siehe Abschnitt 3.3.4 und [58, 28]) führt eine Verfeinerung mit einem lokalen Optimierer jedoch meist nicht zu einer deutlichen Verbesserung. Auch der Einsatz der *Ersatzmodellssuche*, die von diesen Verfahren abgeleitet ist, führt nicht zu einer Steigerung der Effektivität. Als lokaler Optimierer<sup>51</sup> sind die adaptiven Ersatzmodellverfahren wegen ihres meist sehr aufwändigen Modelltrainings und den damit verbundenen hohen Rechenzeiten ebenfalls weniger gut geeignet. Deshalb wird der Einsatz dieser Verfahren hier nicht weiter verfolgt.

### 4.5.1. Überblick und Struktur

Die Verfahrensstruktur ist in Abbildung 4.5 und Algorithmus 4.4 dargestellt. Der Kern des Verfahrens ist der reellwertige Genetische Algorithmus *NSGA-II*<sup>52</sup>.

Nach einer Anzahl von Generationen wird ein Ersatzmodell mit allen verfügbaren Daten erzeugt und dessen Güte überprüft<sup>53</sup>, indem das Bestimmtheitsmaß berechnet und bewertet wird. Ist die Qualität des Modells ausreichend, wird ein Schritt *Ersatzmodellssuche* ausgeführt und im Anschluss daran das Optimum (EOA) oder eine Auswahl von Individuen der PARETO-Menge (MOA) nachgerechnet und zur aktuellen Population hinzugefügt.

Anschließend wird anhand der letzten Generationen geprüft, ob es erfolgversprechend ist, zum lokalen Optimierer zu wechseln<sup>54</sup>. Sind diese Kriterien (siehe Abschnitt 4.2.1) erfüllt, wird die *lokale Suche* mittels *NELDER-MEAD-Verfahren* mit Werten aus der letzten Population gestartet. Wie bei der *Ersatzmodellssuche* werden die Ergebnisse zur aktuellen Population hinzugefügt. Sind die Konvergenzkriterien der *globalen Suche* noch nicht erfüllt, beginnt die nächste Iteration.

Nach Abschluss der Optimierung wird noch einmal ein Ersatzmodell erzeugt und bei ausreichender Güte eine Ersatzmodellssuche durchgeführt<sup>55</sup>.

---

<sup>46</sup>siehe [58, 3]

<sup>47</sup>siehe [49]

<sup>48</sup>siehe Abschnitt 3.2.2.2 und [55, 35]

<sup>49</sup>z.B. SQP

<sup>50</sup>die neuen Punkte sind bereits Lösungen einer Optimierungsaufgabe

<sup>51</sup>z.B. durch eine Ersatzzielfunktion und geeignete Intervallanpassungen

<sup>52</sup>siehe Abschnitt 3.3.3.1 und [12, 53]

<sup>53</sup>siehe 4.3.2.1

<sup>54</sup>siehe 4.2.1

<sup>55</sup>siehe Abschnitt 4.3.2.2

4.5. Das hybride Optimierungsverfahren AHMOS

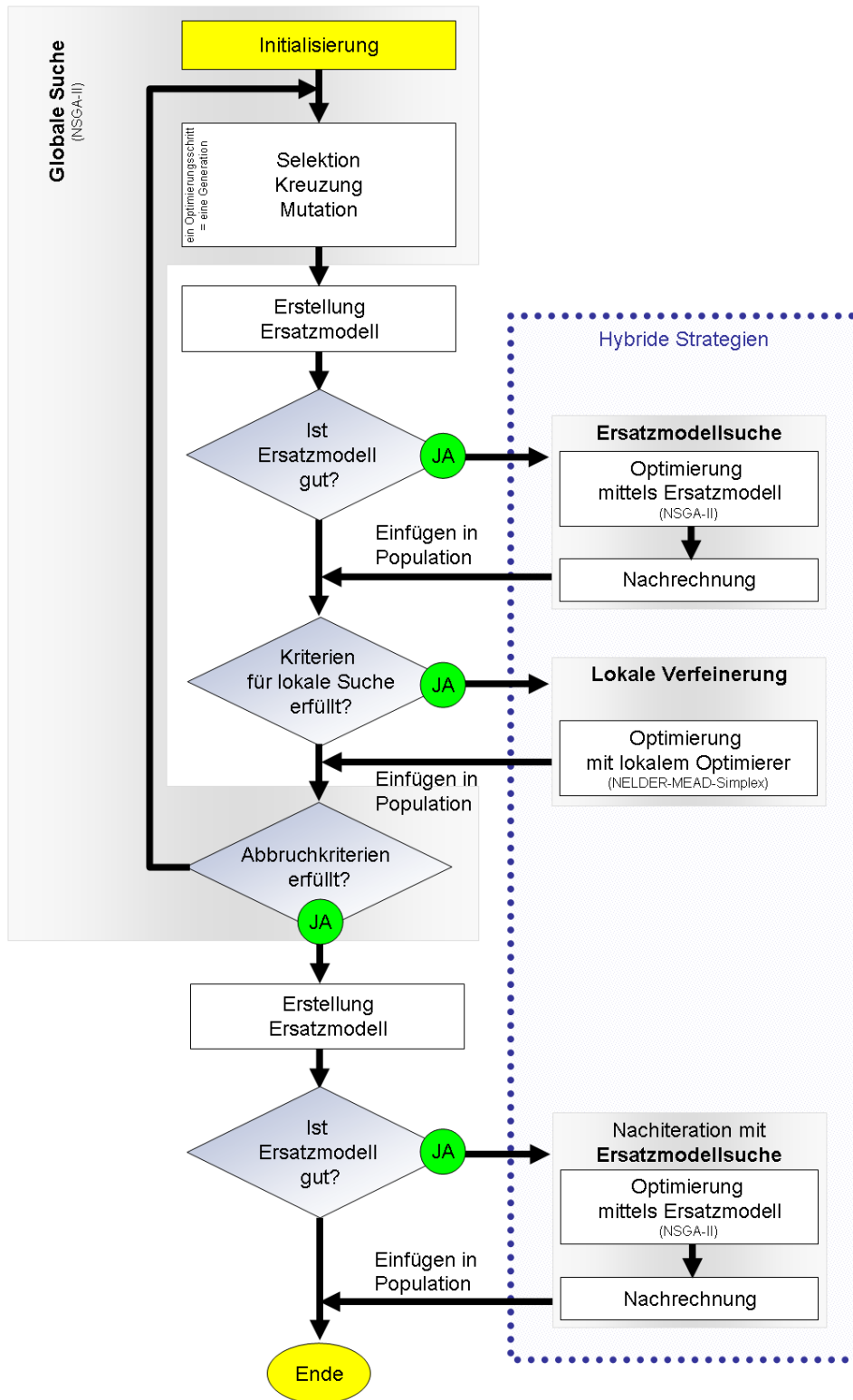


Abbildung 4.5.: Struktur des hybriden Optimierungsverfahrens AHMOS

---

**Algorithmus 4.4** Hybrides Verfahren AHMOS

---

**Eingabe** : OA

Initialisierung

```
while Abbruchbedingungen nicht erfüllt do
  Führe einen globalen Suchschritt aus (eine Generation NSGA-II)
  Erzeuge Ersatzmodell (Moving-Least-Squares)
  if Modell ist gut:  $R^2 > \varepsilon_{R^2}$  then
    Optimiere mit Ersatzmodell
    Nachrechnung mit Original-Zielfunktionen
    Einfügen der Lösungen in Population der Globalen Suche

  if Kriterien für lokale Suche erfüllt then
    Optimiere mit NELDER-MEAD-Simplex
    Einfügen der Lösungen in Population der Globalen Suche

end

Erzeuge Ersatzmodell (Moving-Least-Squares)
if Modell ist gut:  $R^2 > \varepsilon_{R^2}^{(ende)}$  then
  Optimiere mit Ersatzmodell
  Nachrechnung mit Original-Zielfunktionen
  Einfügen der Lösungen in Population der Globalen Suche
```

---

#### 4.5.2. Parallelisierung

Wie bereits in Abschnitt 2.2.2 beschrieben, spielt effektive Parallelisierung eine sehr wichtige Rolle beim Einsatz von Optimierungsmethoden für Aufgabenstellungen mit aufwändigen Funktionsauswertungen. In dem beschriebenen Verfahren AHMOS wird die Parallelisierung für die einzelnen Teilschritte separat betrachtet.

**Globale Suche mit Genetischem Algorithmus NSGA-II:** Dieses Verfahren ist leicht zu parallelisieren, da alle Individuen einer Generation unabhängig voneinander berechnet werden können<sup>56</sup>. Dabei ist aber auf eine sinnvolle Abstimmung der Größe der Population auf die verfügbaren Rechenkerne zu achten. Stehen beispielsweise 18 CPU zur Verfügung, wäre es demnach ungünstig, eine Populationsgröße zu wählen, bei der 20 Individuen pro Generation neu berechnet werden müssen, da sich im letzten Paket ein Großteil der CPUs im Leerlauf befinden würden.

---

<sup>56</sup>d.h. müssen in einer Generation 40 neue Individuen berechnet werden, können die Berechnungen auf 40 CPU verteilt parallel gerechnet werden



**Lokale Suche mit Nelder-Mead:** Das Verfahren für sich betrachtet ist nur sehr begrenzt parallelisierbar, da die Funktionswertberechnung in den Konstruktionsoperationen Reflektion, Expansion und Kontraktion<sup>57</sup> nacheinander berechnet werden müssen. Da es bei diesem Verfahren aber sinnvoll ist, mit verschiedenen Simplexes zu starten, können mehrere Durchläufe parallel bearbeitet werden. Ähnlich verhält es sich auch bei MOA, da hierbei unterschiedliche Stellen der PARETO-Front betrachtet und mit verschiedenen Ersatzzielfunktionen berechnet werden müssen, so dass mehrere lokale Suchen gleichzeitig durchgeführt werden können.

**Ersatzmodellssuche:** Die Modellerstellung und Optimierung auf dem Ersatzmodell<sup>58</sup> benötigen nur sehr wenig Rechenzeit, so dass sich eine Parallelisierung nicht lohnt. Die Nachrechnung der ausgewählten Varianten kann aber parallelisiert ausgeführt werden. Dabei sollte die Anzahl dieser Varianten an die mögliche Parallelisierung angepasst werden.

#### 4.5.3. Ausblick

Durch den modularen Aufbau des Verfahrens ist es leicht möglich, weitere hybride Methoden zu integrieren. Die im Folgenden aufgeführten Strategien sind dabei besonders vielversprechend:

**Adaptive Verfahrensparameteranpassung:** Wie von YOUNGSU YUN in [64] beschrieben, lässt sich die Effektivität der globalen Suche erhöhen, wenn die Verfahrensparameter des Optimierungsalgorithmus im Laufe der Optimierung adaptiv angepasst werden.

**Insel-Modelle:** Besonders bei Aufgaben, deren Lösungen sich in sehr unterschiedlichen Gebieten befinden, bietet es sich an, die Parameterbereiche bei der globalen Suche in disjunkte Gebiete (Inseln) zu zerlegen und voneinander unabhängig zu optimieren<sup>59</sup>.

**Robuste Optimierung:** Bei Aufgabenstellungen für technische Anwendungen wird oft die Forderung aufgestellt, möglichst robuste Lösungen zu finden, so dass das gefundene robuste Minimum nicht unbedingt auch das Globale ist, sondern ein lokales Minimum sein kann, welches aber hinsichtlich seiner Robustheit besser ist als das Globale. Die Berechnung und Bewertung der Robustheit kann als Baustein in das hybride Verfahren integriert werden, indem während der Optimierung eine Robustheitsanalyse durchgeführt und deren Ergebnis in die Fitnessbewertung integriert wird.

**Einsatz anderer Ersatzmodelle:** Für die Nachiteration mit Ersatzmodell kann der Einsatz eines rechenzeitaufwändigeren, dafür aber leistungsfähigeren Modells (z.B. Kriging<sup>60</sup>) von Vorteil sein.

---

<sup>57</sup> siehe Abschnitt 3.2.2.1

<sup>58</sup> siehe Abschnitt 4.3

<sup>59</sup> siehe u.a. [54]

<sup>60</sup> siehe z.B. [58]

#### 4. *Hybride Strategien*

**Andere Verfahren als globale und lokale Optimierer:** Die eingesetzten Optimierungsverfahren für die globalen und lokalen Suchschritte der Optimierung können sehr leicht durch weitere Optimierungsverfahren ersetzt werden, so dass beispielsweise andere Evolutionäre Verfahren oder Partikelschwarmoptimierung für die globale Suche oder die Ersatzmodellensuche verwendet werden können, die für spezielle Aufgabenstellungen besser geeignet sind als der NSGA-II. Für die lokale Suche ist die Untersuchung gradientenbasierter Optimierungsverfahren von Interesse.

**Multikriterielle lokale Optimierer:** Alternativ zur Verwendung einkriterieller Optimierungsverfahren mit Ersatzaufgaben als lokale Optimierer sollte für MOA der Einsatz eines lokal arbeitenden multikriteriellen Optimierers untersucht werden.

# 5

## Kapitel 5.

---

# Benchmark der Optimierungsverfahren

Um die Effektivität des in Abschnitt 4.5 hergeleiteten hybriden Verfahrens AHMOS zu bewerten, wird es mit anderen Optimierungsverfahren verglichen. Dabei kommen die folgenden Vergleichsverfahren zum Einsatz. Für jeden der Vergleichsalgorithmen wird ein Anspruch definiert, dessen Erfüllung durch die hybride Strategie anhand geeigneter Tests nachgewiesen werden soll.

**Monte-Carlo-Methode:** Diese einfache Methode<sup>1</sup> gilt als untere Benchmark-Schranke. Ein effektives Verfahren muss im Mittel stets besser sein.

**Genetischer Algorithmus NSGA-II:** Dieses Optimierungsverfahren<sup>2</sup> wird als globaler Optimierer eingesetzt. Das darauf aufbauende hybride Verfahren AHMOS muss besser sein als die globale Suche für sich allein betrachtet.

**Nelder-Mead-Verfahren:** Dieses Optimierungsverfahren<sup>3</sup> wird als lokaler Optimierer eingesetzt. Das hybride Verfahren AHMOS muss auch hier besser sein als die lokale Suche, die von zufälligen Startpunkten aus gestartet wird.

Eine Vielzahl von Testaufgaben wurden entwickelt, um Optimierungsalgorithmen hinsichtlich ihrer Eignung zum Lösen verschiedener Aufgabenklassen bewerten und vergleichen zu können. Wie bereits in Kapitel 3 dargestellt, gibt es kein allgemein bestes Verfahren, so dass die Beurteilung, ob ein Verfahren gut ist oder nicht, nur in Hinblick auf eine spezielle Aufgabenstellung gegeben werden kann. Bei der Auswahl von Testaufgaben wurde sich daran orientiert, welche Eigenschaften für Zielfunktionen bei aufwändigen technischen Aufgabenstellungen typisch sind.

---

<sup>1</sup>siehe Abschnitt 3.3.2

<sup>2</sup>siehe Abschnitt 3.3.3.1

<sup>3</sup>siehe Abschnitt 3.2.2.1

## 5.1. Test-Aufgaben für einkriterielle Optimierungsaufgaben

Der Erfolg eines Optimierungsverfahrens bei der Lösung einer EOA lässt sich leicht daran messen, wie gut oder wie schnell das globale Optimum gefunden wird. Bei stochastischen Optimierungsverfahren ist es notwendig, die Optimierung mehrmals durchzuführen und so die mittlere Güte und die Streuung zu ermitteln. In den nachfolgenden Tests wird die Qualität des Optimierungsergebnisses daran gemessen, wie gut die bekannten Optima erreicht wurden.

Um eine Vergleichbarkeit herzustellen, wird in den folgenden Tests für alle Optimierungsverfahren eine maximale Anzahl von 300 Funktionsauswertungen vorgegeben. Die Parametrisierung<sup>4</sup> der einzelnen Verfahren ist in Anhang C aufgelistet. Jede Aufgabe wird 250 mal mit identischen Anfangsbedingungen optimiert und ausgewertet. Untersucht wird das Erreichen des optimalen Funktionswertes und (wenn eindeutig) die Entfernung des gefundenen von dem tatsächlichen Optimum nach den 300 Funktionsauswertungen. Die Ergebnisse sind jeweils in einem Verteilungsplot<sup>5</sup> dargestellt. Weiterhin wurde für jeden der 250 Optimierungsläufe getestet, welches der Verfahren die beste Lösung<sup>6</sup> erzielt hat. Diese Ergebnisse werden anschaulich in einer Tortengrafik dargestellt.

### 5.1.1. Testfunktion: Rastrigin

#### 5.1.1.1. Definition

Die RASTRIGIN-Funktion<sup>7</sup> ist eine sehr multimodale Testfunktion. Sie ist charakterisiert durch viele lokale Minima. Für das zu testende Optimierungsverfahren besteht die Schwierigkeit darin, das richtige Tal zu finden, in dem sich das globale Minimum befindet. Häufig wird eine Variante dieser Funktion<sup>8</sup> verwendet, die im Koordinatenursprung  $\mathbf{x}^* = (0, \dots, 0)^T$  ihr Optimum mit dem Wert  $f(\mathbf{x}^*) = 0$  besitzt. Aus numerischen Gründen empfiehlt es sich aber, die Funktion dahingehend zu erweitern, dass sich das Optimum an einer anderen Stelle befindet und einen Wert ungleich Null hat. Die verschobene,  $n$ -dimensionale RASTRIGIN-Funktion für  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{x} = (x_1, \dots, x_n)^T$  ist definiert als

$$f_{\text{Rastrigin}} \begin{cases} 1 + 10n + \sum_{i=1}^n (x_i - 1)^2 - 10 \cos(2\pi(x_i - 1)) \rightarrow \min \\ -5 \leq x_i \leq 5 \quad i = 1, 2, \dots, n \\ \mathbf{x} \in \mathbb{R}^n . \end{cases} \quad (5.1)$$

<sup>4</sup>Verfahrensparameter

<sup>5</sup>Erläuterung des Verteilungsplots: siehe Anhang B

<sup>6</sup>hinsichtlich des Erreichens des optimalen Zielfunktionswertes und der Entfernung vom tatsächlichen globalen Optimums

<sup>7</sup>siehe [60, 40, 58]

<sup>8</sup>siehe z.B. [58, 49]

## 5.1. Test-Aufgaben für einkriterielle Optimierungsaufgaben

Das globale Minimum befindet sich im Punkt  $\mathbf{x}^* \in \mathbb{R}^n$ ,  $\mathbf{x}^* = (1, \dots, 1)^T$  mit dem Wert  $f(\mathbf{x}^*) = 1$ . Als Beispiel ist die 2-dimensionale RASTRIGIN-Funktion<sup>9</sup> in Abbildung 5.1 dargestellt.

### 5.1.1.2. Zielstellung

Mit dieser Testfunktion wird die Fähigkeit des Optimierungsverfahrens getestet, das globale Optimum zu finden, welches von sehr vielen lokalen Optima umgeben ist, die zum Teil nur sehr kleine Verschlechterungen gegenüber dem globalen Optimum aufweisen.

### 5.1.1.3. Test und Auswertung

In den Abbildungen 5.2, 5.4 und 5.6 sind die Ergebnisse für die RASTRIGIN-Testfunktion mit unterschiedlichen Anzahlen von Variablen dargestellt. Gut zu erkennen ist, dass das lokal arbeitende Verfahren NELDER-MEAD mit zufälligen Startpunkten schlechte Ergebnisse liefert. Das ist darin begründet, dass es immer zum nächstliegenden Tal<sup>10</sup> des Startpunktes konvergiert und nicht zum globalen Minimum. Das hybride Verfahren AHMOS liefert im Mittel besserer Ergebnisse als der Genetische Algorithmus NSGA-II und die Streubreite ist ebenfalls kleiner. Die Vorteile werden aber etwas geringer bei steigender Variablenanzahl (siehe Abbildungen 5.4 und 5.6).

Im Verteilungsplot der RASTRIGIN-Funktion mit 2 Variablen ist beim NSGA-II und beim AHMOS anhand der Häufungen sehr gut zu erkennen, dass der Optimierer oft das globale Minimum gefunden hat, jedoch auch häufig in einem der umliegenden lokalen Minima hängen geblieben ist, wobei es dem AHMOS besser gelingt, das globale Optimum zu erreichen.

Im Verfahren AHMOS kommt vorrangig die hybride Strategie „Globale Suche - Lokale Verfeinerung“ zum Einsatz, wobei das lokale Optimum, welches der globale Optimierer gefunden hat, noch verbessert wird. Die *Ersatzmodellsuche* bringt (insbesondere bei vielen Variablen) hierbei nur wenig Vorteil, da die starke Multimodalität mit Hilfe des Ersatzmodells schwer abgebildet werden kann.

In den Abbildungen 5.3, 5.5 und 5.7 ist dargestellt, in welchem Verhältnis die verglichenen Verfahren die besten Lösungen ermittelt haben. Dabei ist zu erkennen, dass auch bei größerer Variablenanzahl das Hybridverfahren Vorteile aufweist.

**Ergebnis:** Das beste Resultat wurde mit dem hybriden Verfahren AHMOS erzielt.

---

<sup>9</sup>(5.1) mit  $n = 2$

<sup>10</sup>lokalem Minimum

5. Benchmark der Optimierungsverfahren

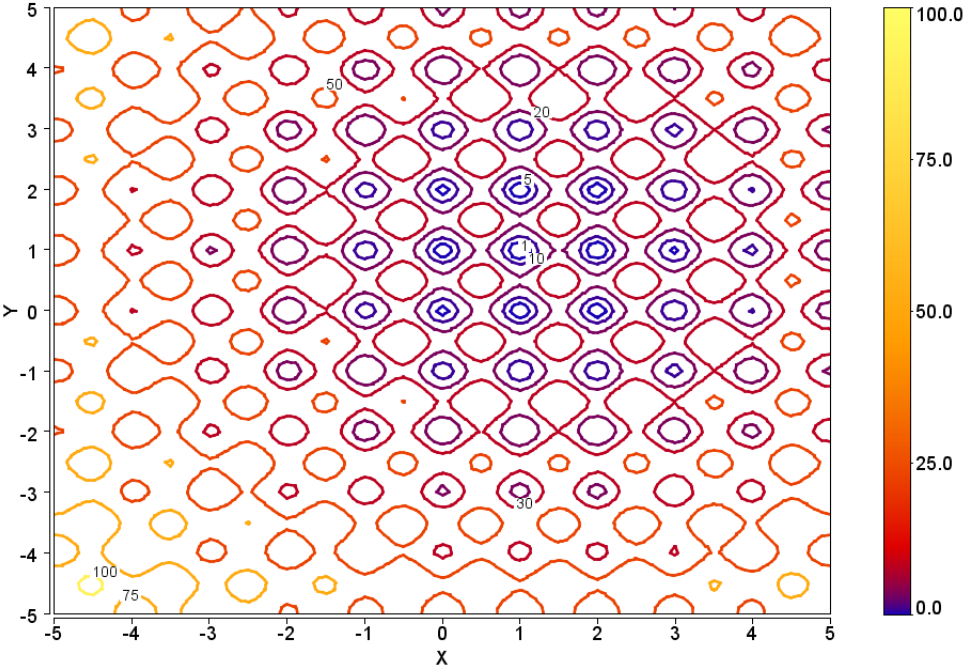
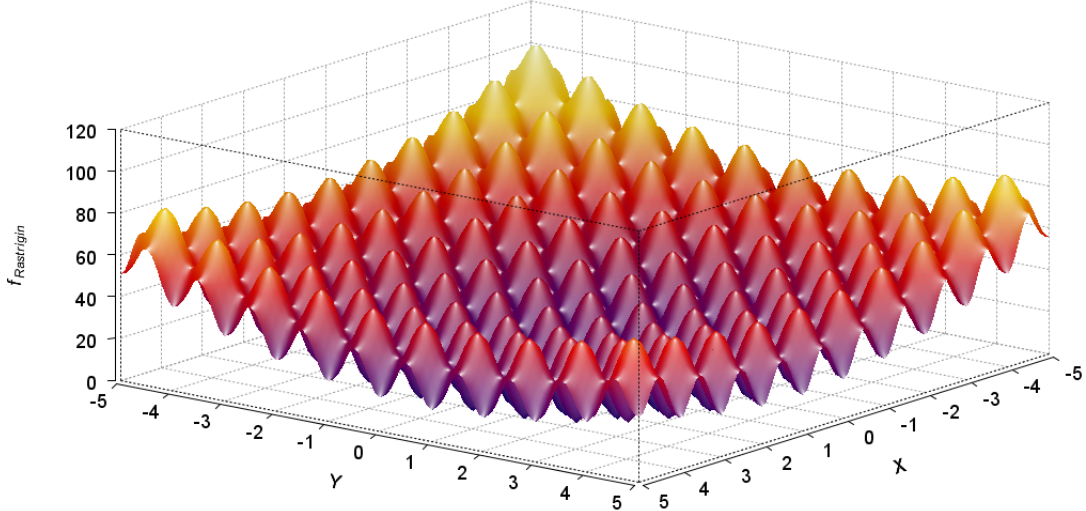


Abbildung 5.1.: Testfunktion: 2-dimensionale RASTRIGIN

### 5.1. Test-Aufgaben für einkriterielle Optimierungsaufgaben

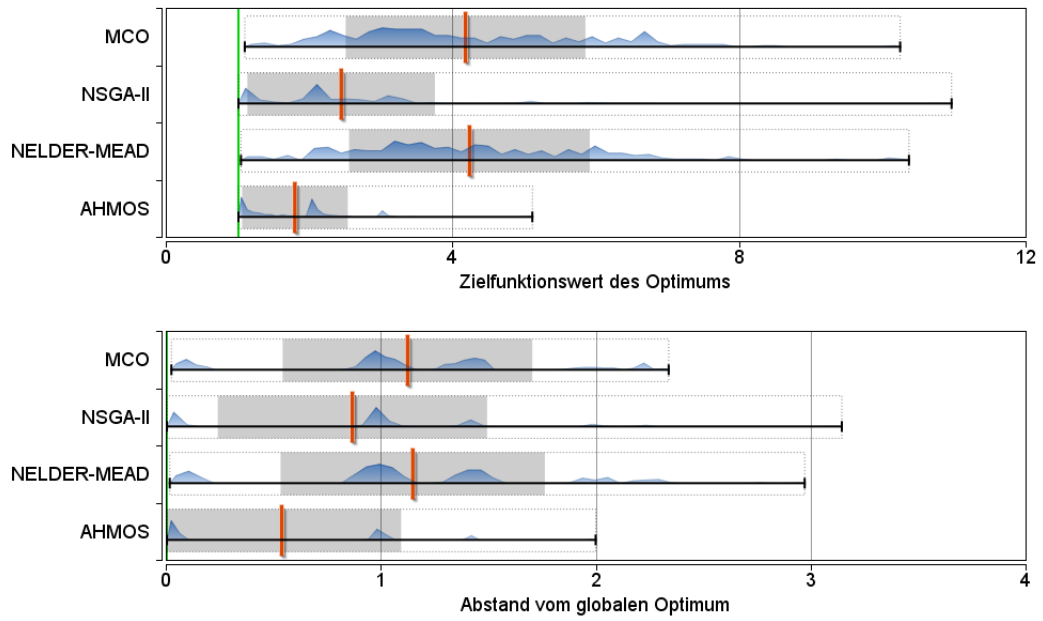


Abbildung 5.2.: Auswertung: RASTRIGIN mit 2 Variablen

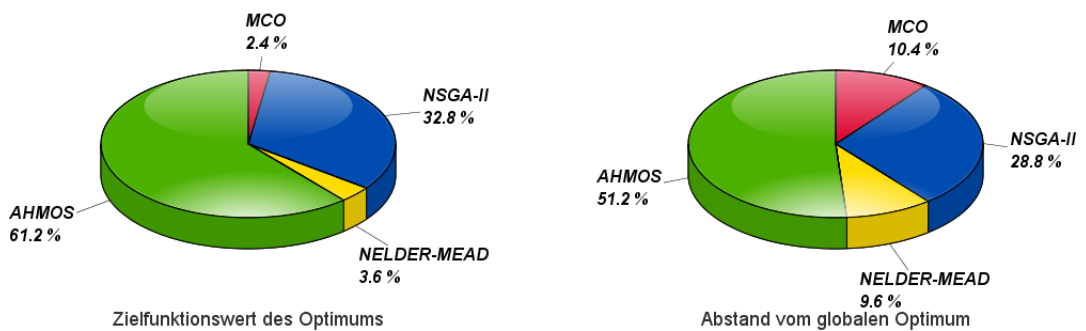


Abbildung 5.3.: Auswertung (bestes Verfahren): RASTRIGIN mit 2 Variablen

## 5. Benchmark der Optimierungsverfahren

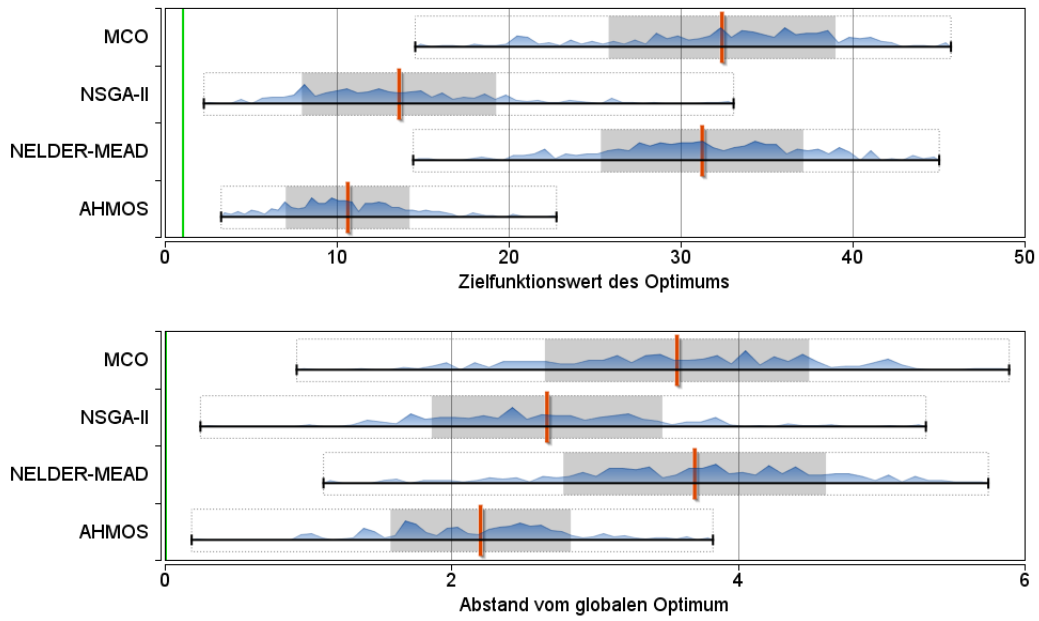


Abbildung 5.4.: Auswertung: RASTRIGIN mit 5 Variablen

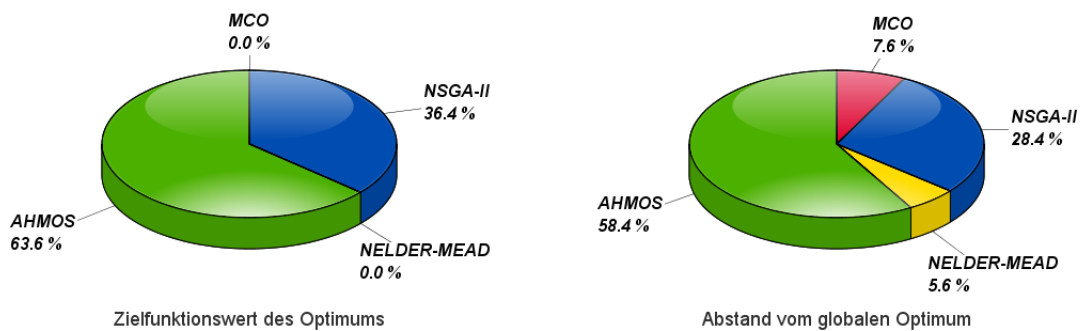


Abbildung 5.5.: Auswertung (bestes Verfahren): RASTRIGIN mit 5 Variablen



5.1. Test-Aufgaben für einkriterielle Optimierungsaufgaben

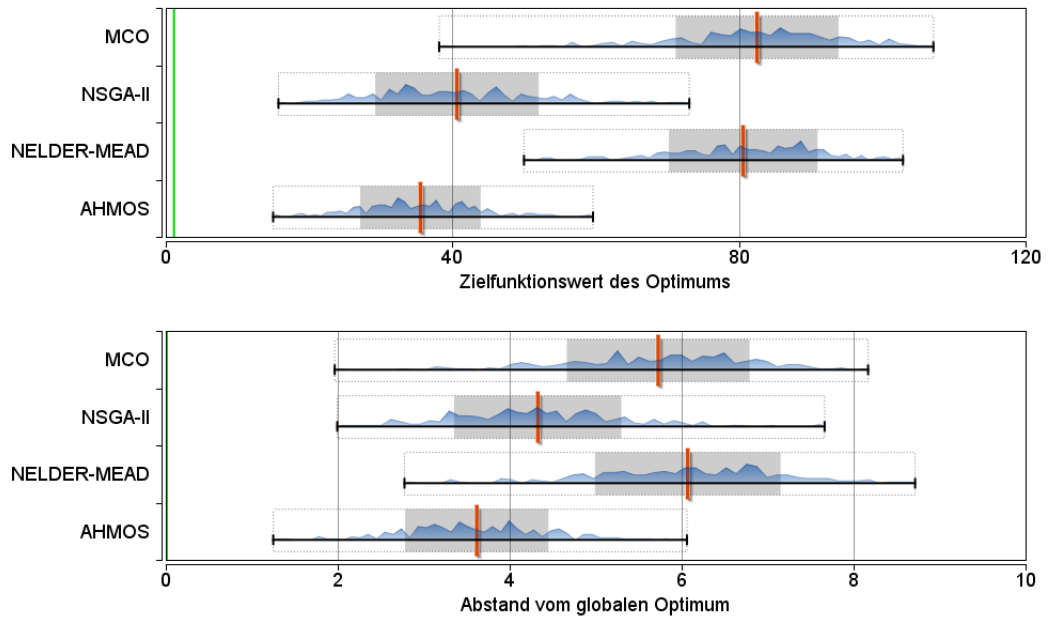


Abbildung 5.6.: Auswertung: RASTRIGIN mit 9 Variablen

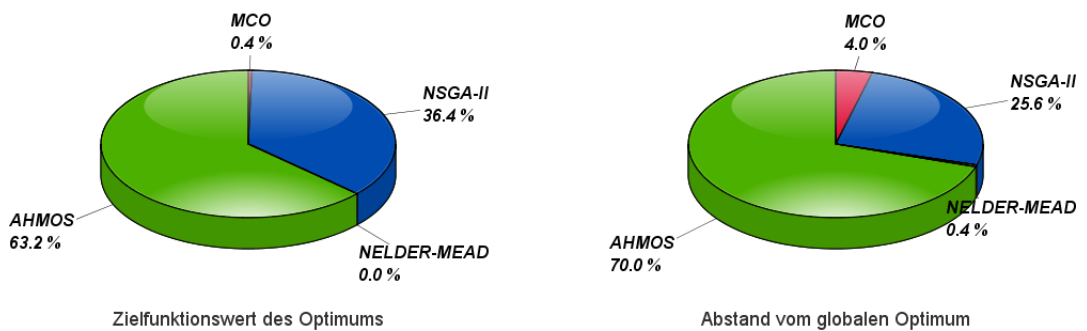


Abbildung 5.7.: Auswertung (bestes Verfahren): RASTRIGIN mit 9 Variablen

## 5. Benchmark der Optimierungsverfahren

### 5.1.2. Testfunktion: Rosenbrock

#### 5.1.2.1. Definition

Die ROSENBRÖCK-Funktion<sup>11</sup> ist durch ein bananenförmiges Tal charakterisiert, in dem sich auch das globale Minimum befindet. Dabei ist es zwar nicht schwer, das Tal zu finden, jedoch erweist sich die Suche nach dem eindeutigen globalen Minimum als schwierig. Das Optimum der Aufgabe befindet sich im Punkt  $\mathbf{x}^* \in \mathbb{R}^n$ ,  $\mathbf{x}^* = (1, \dots, 1)^T$  mit dem Funktionswert  $f(\mathbf{x}^*) = 0$ . Die  $n$ -dimensionale ROSENBRÖCK-Funktion für  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{x} = (x_1, \dots, x_n)^T$  ist definiert als

$$f_{\text{Rosenbrock}} \begin{cases} \sum_{i=1}^{n-1} [(1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2] \rightarrow \min \\ -5 \leq x_i \leq 5 \quad i = 1, 2, \dots, n \\ \mathbf{x} \in \mathbb{R}^n . \end{cases} \quad (5.2)$$

Als Beispiel ist die 2-dimensionale ROSENBRÖCK-Funktion<sup>12</sup> in Abbildung 5.8 dargestellt.

#### 5.1.2.2. Zielstellung

Mit dieser Testfunktion wird die Fähigkeit des Optimierungsverfahrens getestet, das globale Optimum zu finden, welches sich in einem langgezogen Tal ohne signifikanten Gradienten befindet.

#### 5.1.2.3. Test und Auswertung

Den Abbildungen 5.9 und 5.11 ist zu entnehmen, dass die Funktion gut mit dem Genetischen Algorithmus NSGA-II und dem hybriden Verfahren AHMOS zu lösen ist. Das Verfahren NELDER-MEAD mit zufälligen Startpunkten liefert hingegen schlechte Ergebnisse. Das ist darin begründet, dass es mit den Operationen<sup>13</sup> des Algorithmus schwer ist, in dem Tal der Funktion mit sehr flacher Talsohle in Richtung des globalen Minimums voranzukommen. In der hybriden Strategie „Globale Suche - Lokale Verfeinerung“, bei der ausgehend von einem guten Punkt, den die globale Optimierung geliefert hat und die recht schnell in die Nähe des Optimums kommt, ist noch eine Verbesserung möglich. Die hybride Strategie „Ersatzmodellssuche“ wirkt dabei unterstützend, um schnell in die Nähe des globalen Minimums zu gelangen.

In den Abbildungen 5.10 und 5.12 ist für die ROSENBRÖCK-Funktion dargestellt, in welchem Verhältnis die verglichenen Verfahren die besten Lösungen ermittelt haben. Bei der Bewertung des Abstandes vom globalen Minimum ist im Vergleich mit dem NSGA-II keine große Verbesserung zu erkennen, währenddessen beim Erreichen des optimalen Funktionswert ein Vorteil

---

<sup>11</sup>siehe [50, 58]

<sup>12</sup>(5.2) mit  $n = 2$

<sup>13</sup>siehe Abschnitt 3.2.2.1

5.1. Test-Aufgaben für einkriterielle Optimierungsaufgaben

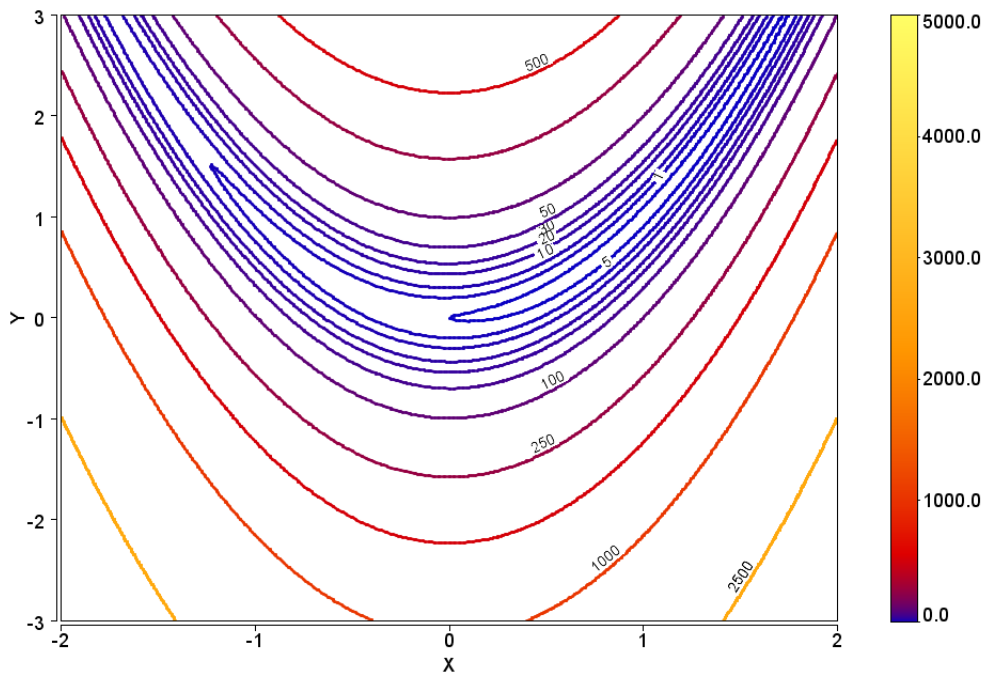
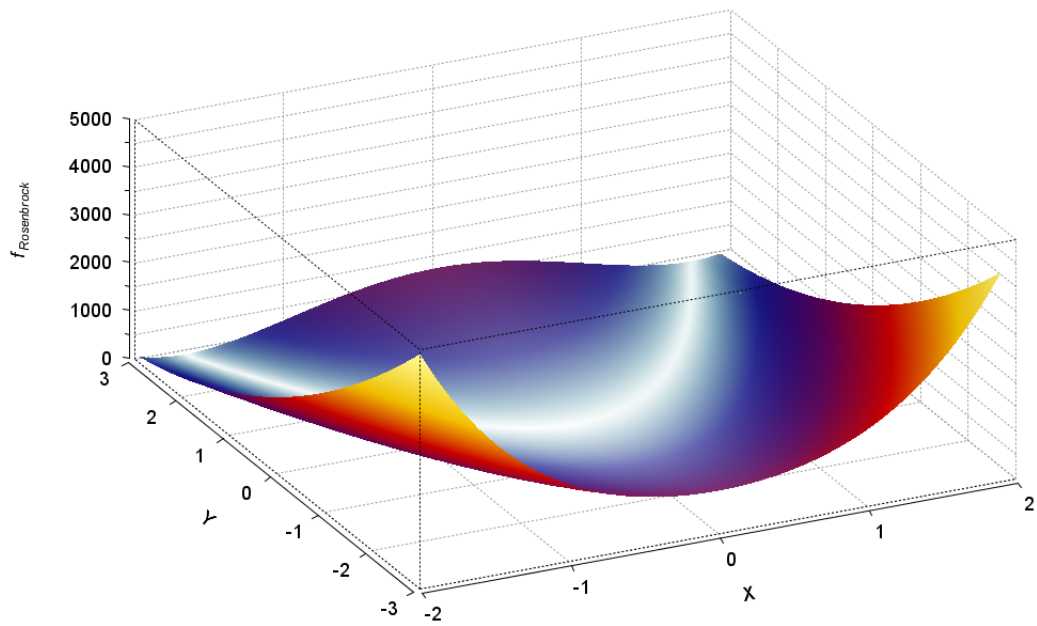


Abbildung 5.8.: Testfunktion: 2-dimensionale ROSENBRÖCK

## 5. Benchmark der Optimierungsverfahren

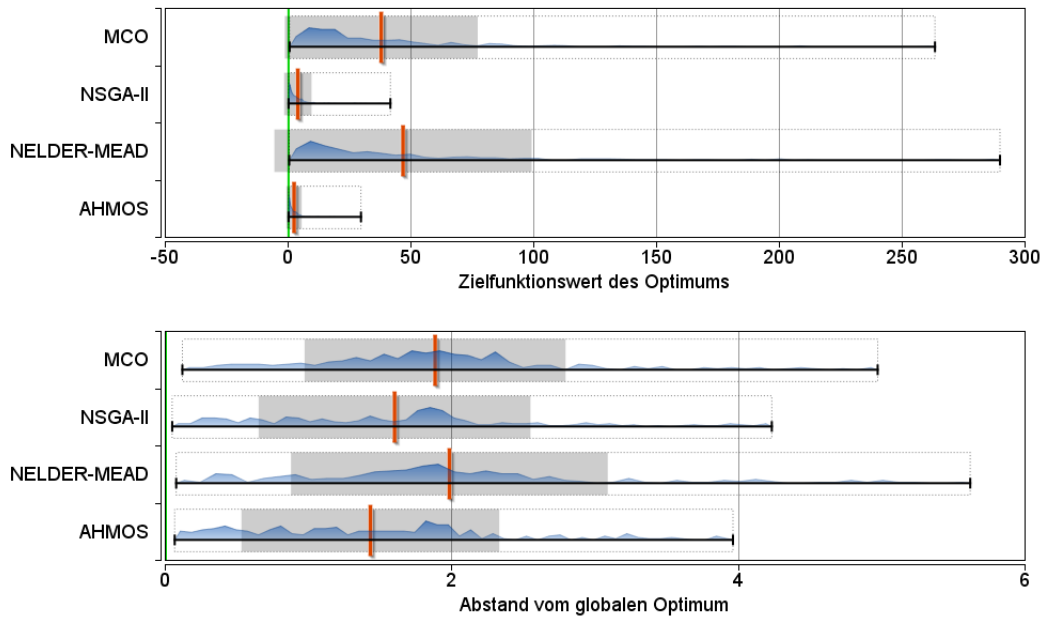


Abbildung 5.9.: Auswertung: ROSENBRACK mit 3 Variablen

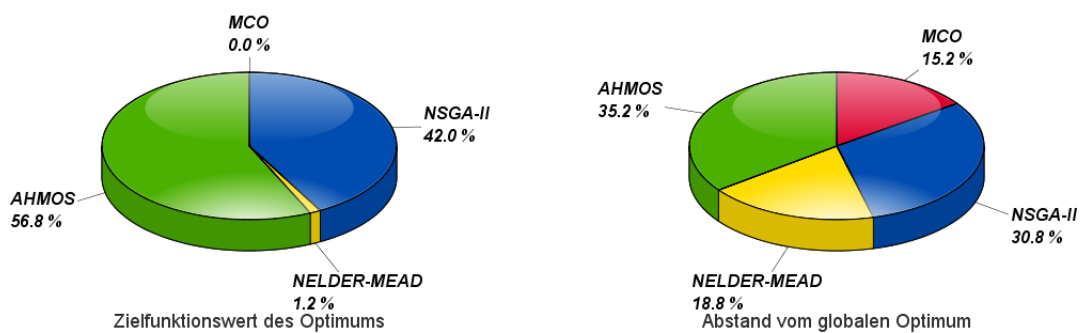


Abbildung 5.10.: Auswertung (bestes Verfahren): ROSENBRACK mit 3 Variablen

### 5.1. Test-Aufgaben für einkriterielle Optimierungsaufgaben

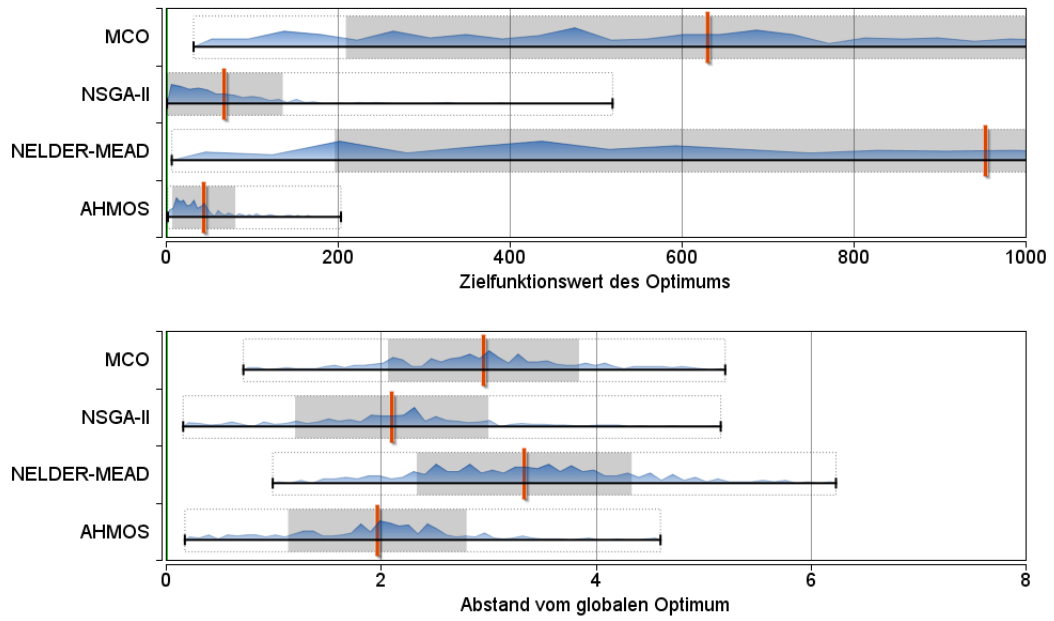


Abbildung 5.11.: Auswertung: ROSENBROCK mit 5 Variablen

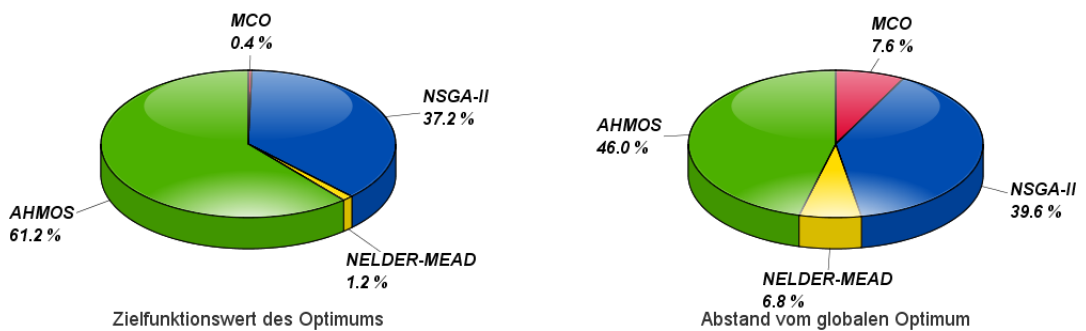


Abbildung 5.12.: Auswertung (bestes Verfahren): ROSENBROCK mit 5 Variablen

## 5. Benchmark der Optimierungsverfahren

zu erkennen ist. Der Nutzen der hybriden Strategien steigt bei dieser Testaufgabe mit höherer Dimension.

**Ergebnis:** Das beste Resultat wurde mit dem hybriden Verfahren AHMOS erzielt, wobei die Verbesserungen im Vergleich zum NSGA-II moderat sind.

### 5.1.3. Testfunktion: Himmelblau

#### 5.1.3.1. Definition

Die HIMMELBLAU-Funktion<sup>14</sup> mit 2 Variablen besitzt vier identische globale Minima, so dass es für das Optimierungsverfahren nicht eindeutig ist, zu welchem Optimum es konvergieren soll. Die Minima befinden sich in den Punkten

$$\begin{aligned} \mathbf{x}_1^* &= (2, 3)^T \\ \mathbf{x}_2^* &= (-2.805118, 3.131312)^T \\ \mathbf{x}_3^* &= (-3.779310, -3.283186)^T \\ \mathbf{x}_4^* &= (3.584428, -1.848126)^T \end{aligned} \quad ,$$

jeweils mit dem Funktionswert  $f(\mathbf{x}_i^*) = 0$ ,  $i = 1, \dots, 4$ . Die HIMMELBLAU-Funktion (siehe Abbildung 5.13) ist definiert als

$$f_{\text{Himmelblau}} \begin{cases} (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7) \rightarrow \min \\ -6 \leq x_i \leq 6 \quad i = 1, 2 \\ \mathbf{x} \in \mathbb{R}^2. \end{cases} \quad (5.3)$$

#### 5.1.3.2. Zielstellung

Mit dieser Testfunktion wird die Fähigkeit des Optimierungsverfahrens getestet, nicht eindeutige globale Optima zu finden.

#### 5.1.3.3. Test und Auswertung

Der Abbildung 5.14 ist zu entnehmen, dass insbesondere beim NELDER-MEAD-Verfahren mit zufälligen Startpunkten die Streuung groß ist. Eine Ursache dafür kann darin begründet sein, dass sich die Eckpunkte des Simplexes des Verfahrens in der Nähe verschiedener Optima befinden und mit den Konstruktionsoperatoren<sup>15</sup> keine Verbesserungen gefunden werden können und somit das Verfahren abbricht.

In Kombination mit dem an sich gut arbeitenden NSGA-II in der hybriden Strategie „Globale Suche - Lokale Verfeinerung“ ist jedoch eine deutliche Reduzierung der Streubreite erzielbar.

<sup>14</sup>siehe [4, 65]

<sup>15</sup>siehe Abschnitt 3.2.2.1

5.1. Test-Aufgaben für einkriterielle Optimierungsaufgaben

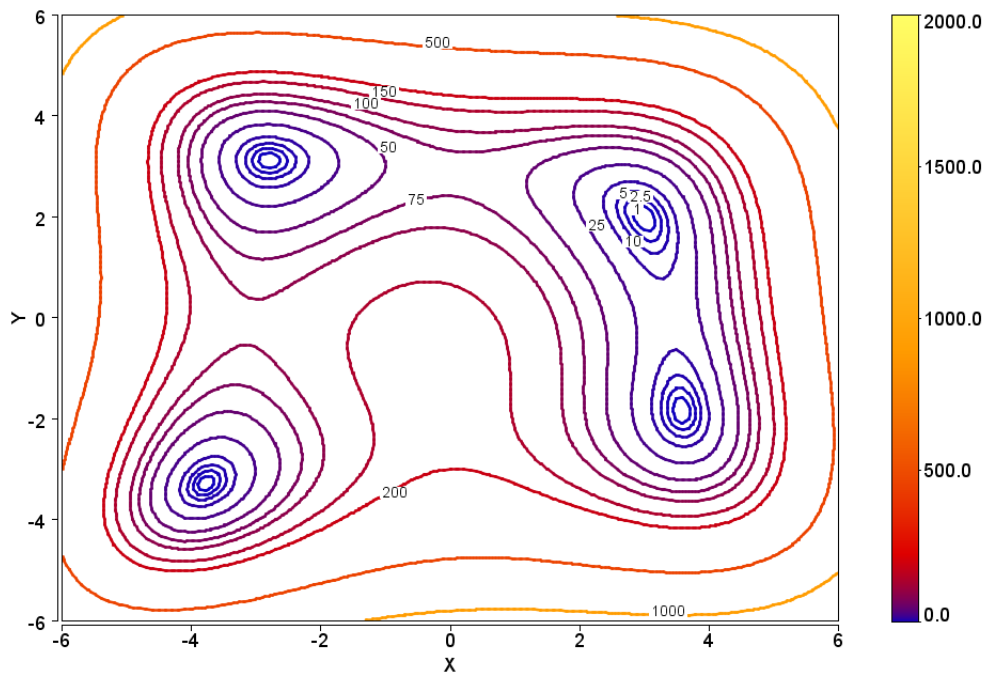
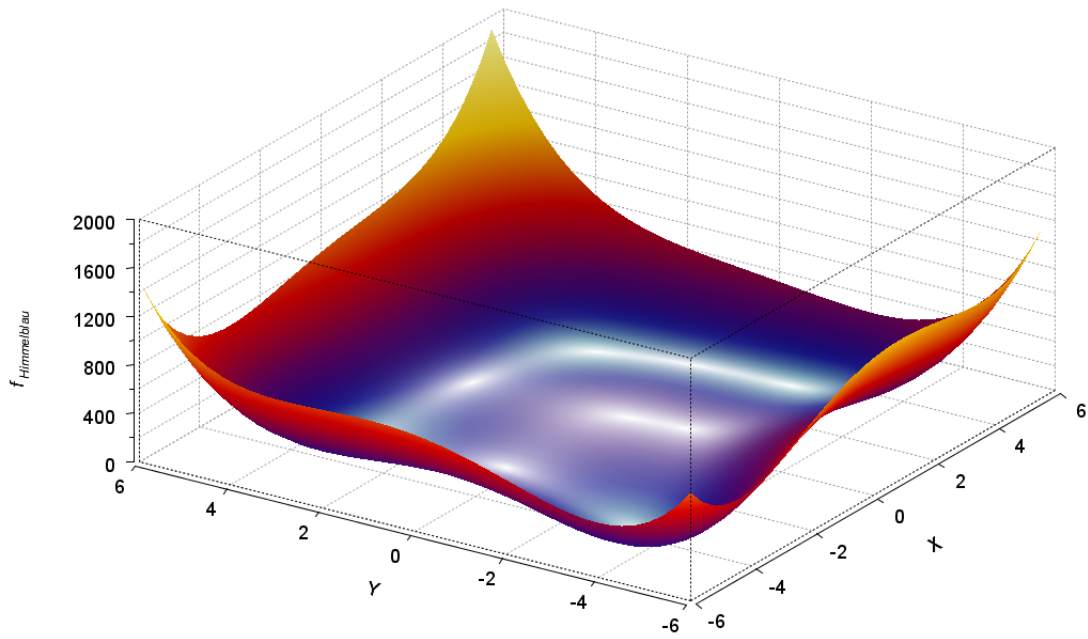


Abbildung 5.13.: Testfunktion: HIMMELBLAU

## 5. Benchmark der Optimierungsverfahren

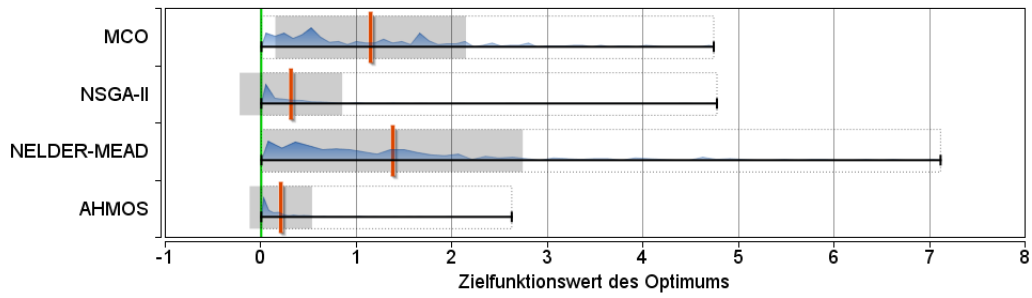


Abbildung 5.14.: Auswertung: HIMMELBLAU (2 Variablen)

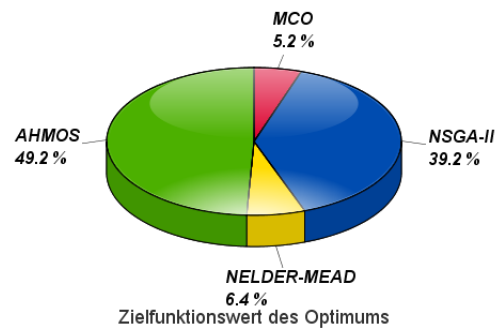


Abbildung 5.15.: Auswertung (bestes Verfahren): HIMMELBLAU (2 Variablen)



Auch die „Ersatzmodellssuche“ kommt zum Einsatz und beschleunigt zum einen die Konvergenz bzw. verbessert noch das gefundene Minimum in der Nachiteration.

In der Abbildung 5.15 ist dargestellt, in welchem Verhältnis die verglichenen Verfahren die besten Lösungen ermittelt haben. Die in Abbildung 5.14 aufgezeigten Vorteile sind in dieser Untersuchung jedoch nicht so gut ablesbar.

Eine Untersuchung des Abstandes vom globalen Minimum fand hier nicht statt, da mehrere unterschiedliche Minima vorhanden sind.

**Ergebnis:** Das beste Resultat wurde mit dem hybriden Verfahren AHMOS erzielt, wobei eine deutliche Verbesserung im Vergleich zum NSGA-II erkennbar ist.

#### 5.1.4. Testfunktion: Deb2

##### 5.1.4.1. Definition

Aufbauend auf der HIMMELBLAU-Funktion wurde von KALYANMOY DEB eine restringierte Optimierungsaufgabe<sup>16</sup> formuliert, bei der die eng begrenzte zulässige Menge durch die Schnittmenge zweier Kreise gegeben ist:

$$f_{Deb2} \begin{cases} (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7) \rightarrow \min \\ x_1^2 + (x_2 - 2.5)^2 \geq 4.84 \\ (x_1 - 0.3)^2 + (x_2 - 2.5)^2 \leq 4.84 \\ 0 \leq x_i \leq 6 \quad i = 1, 2 \\ \mathbf{x} \in \mathbb{R}^2 . \end{cases} \quad (5.4)$$

Für das Optimierungsverfahren besteht die Herausforderung zunächst darin, zulässige Punkte in dem schmalen, nichtlinearen Restriktionsbereich zu finden. Das Optimum liegt bei  $\mathbf{x}^* \in \mathbb{R}^2$ ,  $\mathbf{x}^* \approx (2.491, 2.299)$  mit dem Funktionswert  $f(\mathbf{x}^*) \approx 6.8324$ .

##### 5.1.4.2. Zielstellung

Mit dieser Testfunktion wird geprüft, wie gut es dem Optimierungsverfahren gelingt, den eng begrenzten zulässigen Bereich und das globale Minimum zu finden, welches auf dem Rand der zulässigen Menge liegt. Weiterhin wird getestet, ob das Optimierungsverfahren korrekt mit Nebenbedingungen umgehen kann. In Abbildung 5.16 ist die Testfunktion dargestellt, wobei der zulässige Bereich hervorgehoben ist.

##### 5.1.4.3. Test und Auswertung

Die Testergebnisse sind in Abbildung 5.17 dargestellt. Besonders auffällig ist, dass das MONTE-CARLO-Verfahren eine deutlich kleinere Streubreite aufweist als alle anderen Verfahren. Das

<sup>16</sup>siehe [11, 58]

5. Benchmark der Optimierungsverfahren

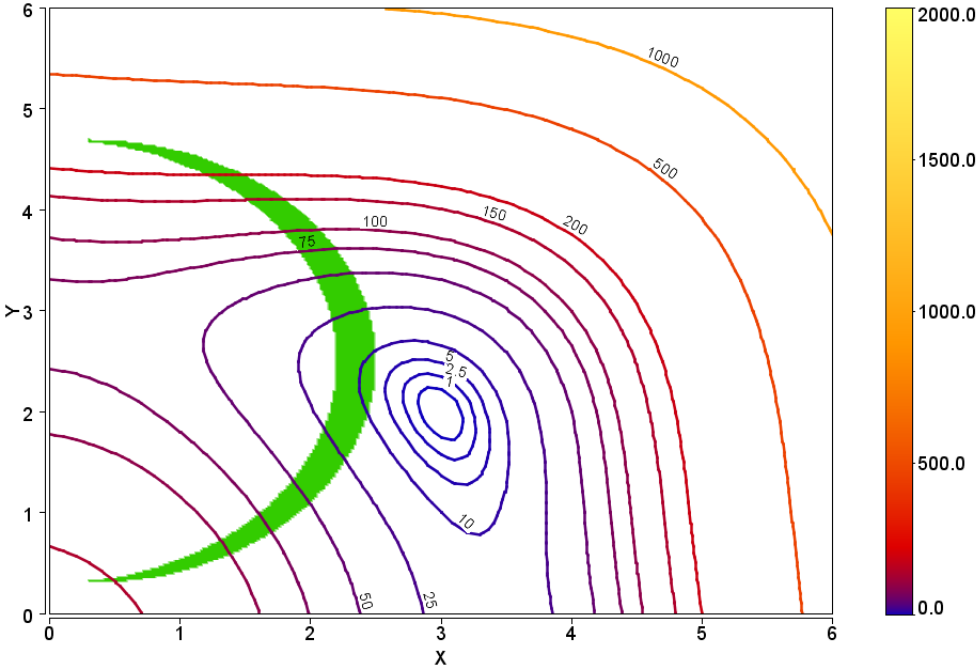
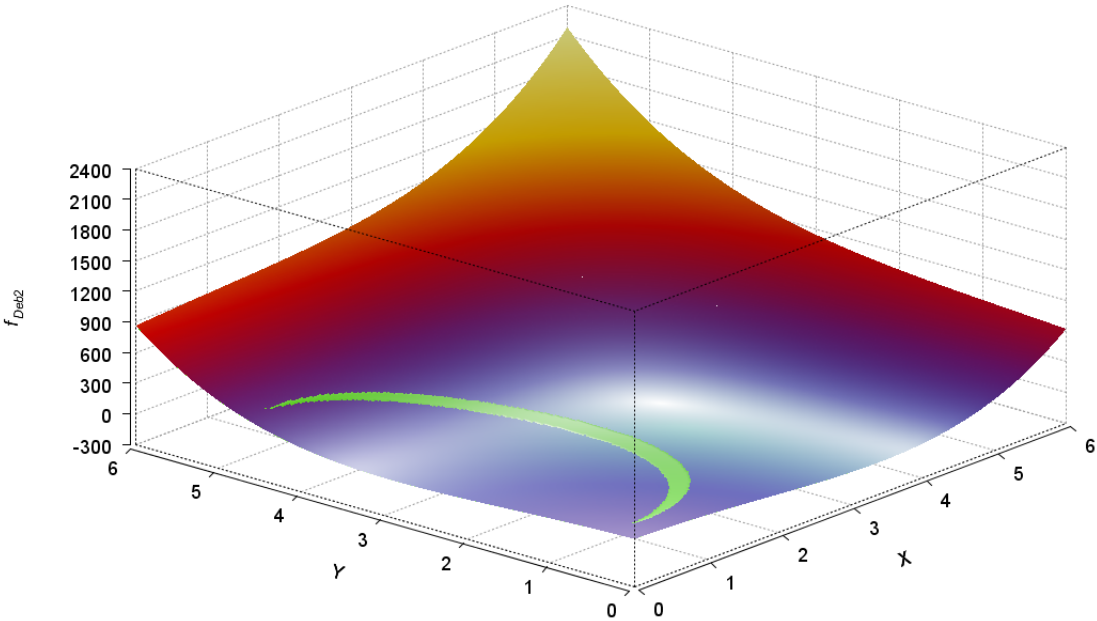


Abbildung 5.16.: Testfunktion: DEB2 (zulässiger Bereich hervorgehoben)

5.1. Test-Aufgaben für einkriterielle Optimierungsaufgaben

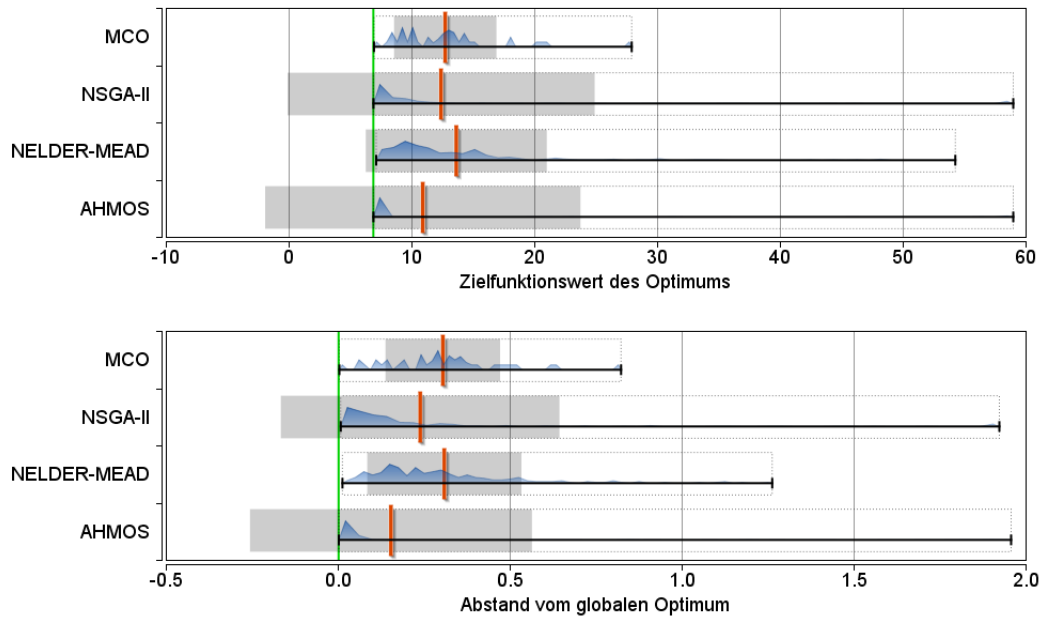


Abbildung 5.17.: Auswertung: DEB2 (2 Variablen)

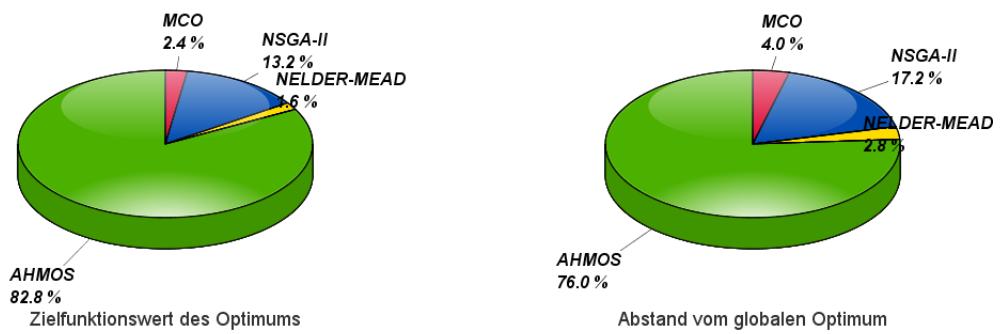


Abbildung 5.18.: Auswertung (bestes Verfahren): DEB2 (2 Variablen)

## 5. Benchmark der Optimierungsverfahren

hybride Verfahren AHMOS ist in den meisten Fällen sehr dicht an das Minimum herangekommen, jedoch ist es dem Verfahren in ca. 6% der Optimierungsläufe nicht gelungen, eine gute Lösung zu finden. Durch diese Ausreißer ist der Streubereich groß und der Mittelwert verschoben. Bei dem Genetischen Algorithmus NSGA-II ist prinzipielle das Gleiche zu erkennen, allerdings wurden dort in ca. 9% der Läufe keine guten Lösungen gefunden.

Durch den Einsatz der hybriden Strategien „Globale Suche - Lokale Verfeinerung“ und „Ersatzmodellsuche“ wird im Vergleich mit dem NSGA-II eine deutliche Verbesserung der Konvergenz erreicht. Die Anzahl der Ausreißer wird ebenfalls reduziert, jedoch nicht vollständig eliminiert.

In Abbildung 5.18, in der aufgezeigt wird, in welchem Verhältnis die verglichenen Verfahren die besten Lösungen ermittelt haben, ist zu erkennen, dass der Einsatz des Hybridverfahrens deutliche Vorteile im Vergleich zu den anderen Verfahren aufweist.

**Ergebnis:** Die im Mittel besten Resultate wurden mit dem hybriden Verfahren AHMOS erzielt, wobei jedoch einzelne Ausreißerlösungen keine gute Lösung für die Testaufgabe darstellen.

### 5.1.5. Testfunktion: Alpental

#### 5.1.5.1. Definition

Die ALPENTAL-Testfunktion<sup>17</sup> ist dadurch charakterisiert, dass sie zwei voneinander getrennte lokale Minima besitzt, von denen eines das globale Minimum ist. Zwischen diesen beiden Tälern befindet sich ein hoher Berg (siehe Abbildung 5.19), der beide Täler voneinander trennt. Zusätzlich wird die mögliche kürzeste Verbindung durch eine Restriktion unterbrochen. Diese Funktion ist geeignet, das in Abschnitt 2.2.1.4 beschriebene „Alpen-Problem“ zu simulieren. Die lokalen Optima sind leicht zu finden. Befindet sich die Suche aber auf der falschen Seite des Berges, ist es zumeist schwierig, auf die andere Seite zu gelangen, um das globale Optimum zu finden.

Die Funktion ist definiert als

$$f_{\text{Alpental}} \left\{ \begin{array}{l} 100 \frac{x_1 x_2}{x_1^2 + x_2^2} + \left( \sum_{i=1}^2 (x_i + (-1)^i)^2 \right) \left( \sum_{i=1}^2 (x_i + (-1)^{i+1})^2 + 1 \right) \rightarrow \min \\ x^2 + y^2 > \frac{1}{4} \\ -5 \leq x_i \leq 5 \quad i = 1, 2 \\ \mathbf{x} \in \mathbb{R}^2 . \end{array} \right. \quad (5.5)$$

Das globale Optimum der Aufgabe befindet sich im Punkt  $\mathbf{x}^* \in \mathbb{R}^2$ ,  $\mathbf{x}^* = (-1, 1)^T$  mit dem Funktionswert  $f(\mathbf{x}^*) = -50$ .

<sup>17</sup>wurde im Rahmen dieser Arbeit auf Basis einer von DEB in [10] vorgestellten Funktion konstruiert

5.1. Test-Aufgaben für einkriterielle Optimierungsaufgaben

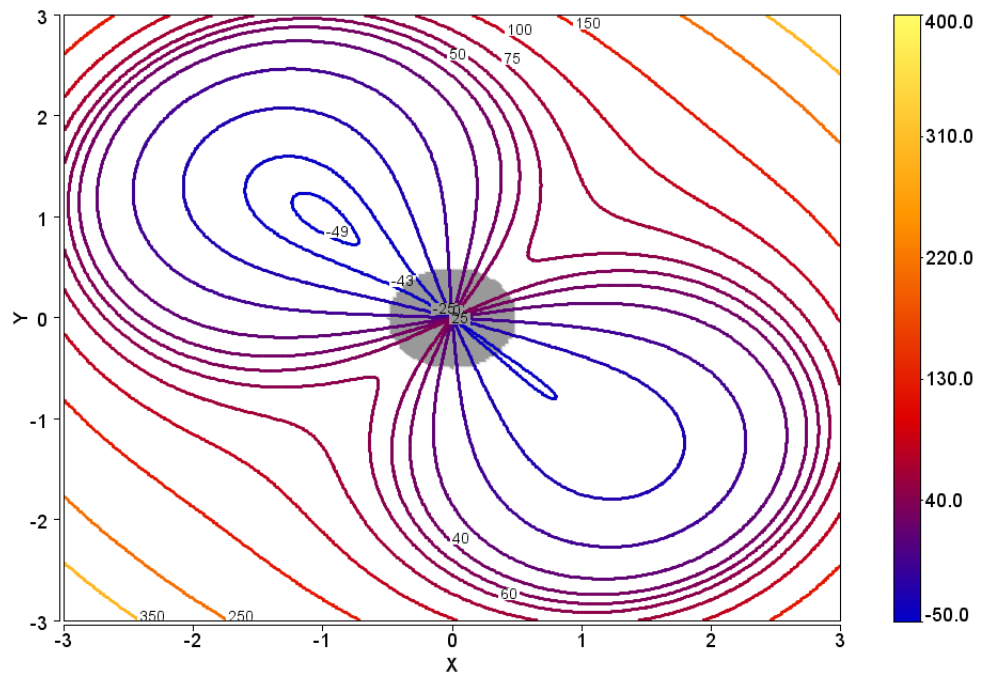
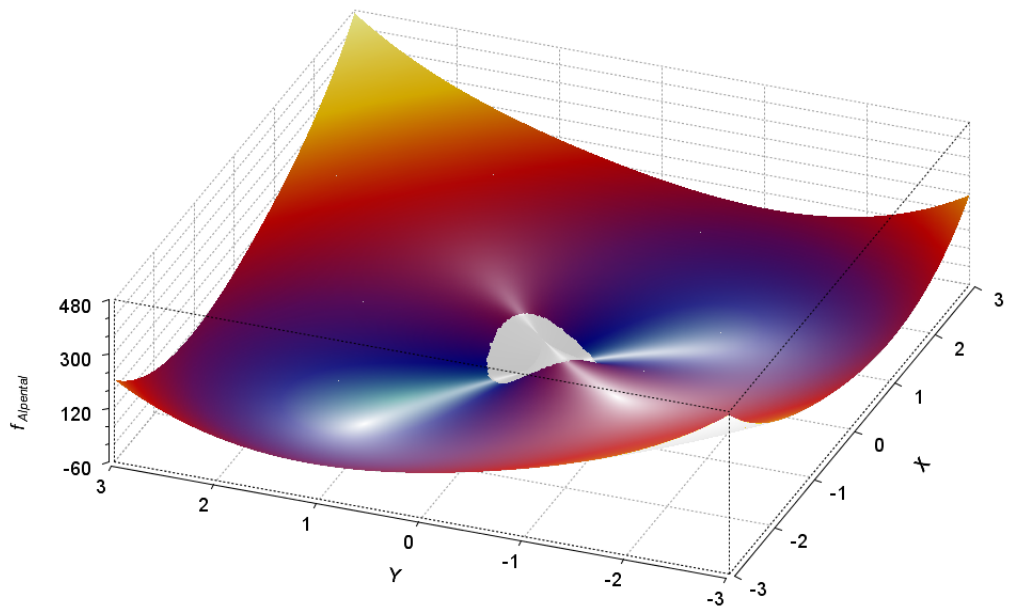


Abbildung 5.19.: Testfunktion: ALPENTAL (Unzulässiger Bereich grau hervorgehoben)

## 5. Benchmark der Optimierungsverfahren

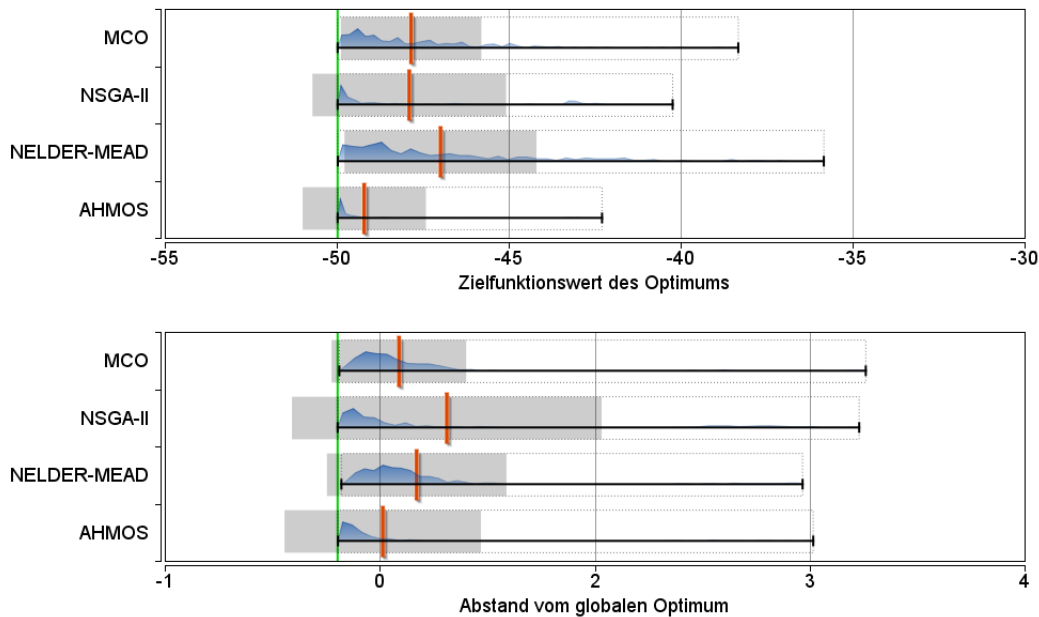


Abbildung 5.20.: Auswertung: ALPENTAL (2 Variablen)

### 5.1.5.2. Zielstellung

Mit der ALPENTAL-Testfunktion kann die Konstellation simuliert und getestet werden, dass mehrere lokale Optima existieren, die durch schlechte Bereiche voneinander getrennt sind, so dass der Optimierer wieder den Berg oder einen unzulässigen Bereich überwinden muss, um die anderen Gebiete untersuchen zu können und das globale Optimum auch dann zu finden, wenn er anfänglich im falschen Tal gesucht hat.

### 5.1.5.3. Test und Auswertung

Die Testergebnisse sind in Abbildung 5.20 dargestellt. Bei dem Genetischen Algorithmus NSGA-II ist zu erkennen, dass die Optimierung in einigen der Durchläufe im Tal des lokalen Minimums stagniert ist, was an der Häufung bei  $f(\mathbf{x}^*) \approx -43$  sichtbar wird. Das NELDER-MEAD-Verfahren mit zufälligen Startpunkten kommt recht schnell in die Nähe eines der beiden lokalen Minima, konvergiert dann aber nur langsam, solange der Ausgangspunkt noch weit vom Optimum entfernt ist.

Im hybriden Verfahren AHMOS kommen beide Strategien „Globale Suche - Lokale Verfeinerung“ und die „Ersatzmodellssuche“ zum Einsatz. Die „Ersatzmodellssuche“ wirkt hierbei insbesondere unterstützend für die Globale Suche, weshalb der AHMOS seltener im Tal des lokalen Minimums stagniert. Außerdem verbessert sie das gefundene Minimum in der Nachiteration. Die Strategie „Globale Suche - Lokale Verfeinerung“ kommt dann erfolgreich zum

## 5.1. Test-Aufgaben für einkriterielle Optimierungsaufgaben

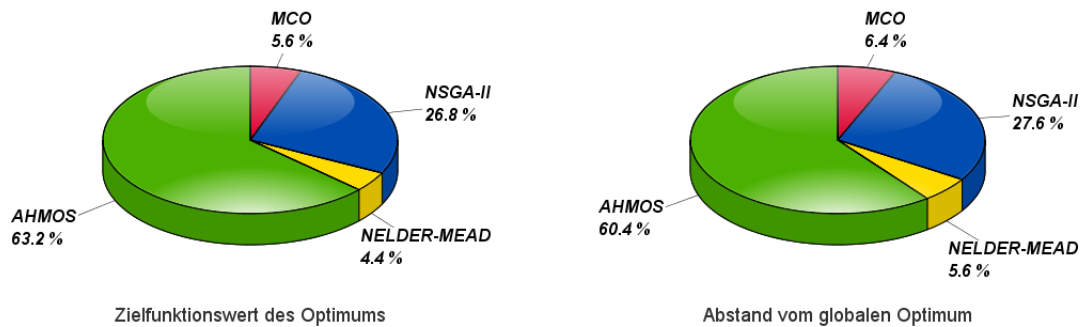


Abbildung 5.21.: Auswertung (bestes Verfahren): ALPENTAL (2 Variablen)

Einsatz, wenn die globale Optimierung und die „Ersatzmodellssuche“ bereits Lösungen in der Nähe der Minima gefunden haben.

In Abbildung 5.21 ist dargestellt, in welchem Verhältnis die verglichenen Verfahren die besten Lösungen ermittelt haben. Klar zu erkennen ist, dass mit den hybriden Strategien das globale Optimum deutlich besser aufgefunden wurde.

**Ergebnis:** Das beste Resultat wurde mit dem hybriden Verfahren AHMOS erzielt, wobei jedoch einige wenige Ausreißer nur das lokale Minimum gefunden haben.

### 5.1.6. Testfunktion: Miner-S-1

#### 5.1.6.1. Definition

Die Testfunktion MINER-S-1<sup>18</sup> ist eine multimodale Funktion mit vielen, unregelmäßig verteilten lokalen Minima, die sich in großen und kleinen Tälern verteilen. Einige dieser Täler bilden große Plateaus, andere sind nur sehr schmal. Das globale Minimum befindet sich im Punkt  $\mathbf{x}^* \in \mathbb{R}^n$ ,  $\mathbf{x}^* = (1, \dots, 1)^T$ , besitzt den Wert  $f(\mathbf{x}^*) = 0$  und befindet sich in einem sehr

<sup>18</sup>diese Testfunktion wurde im Rahmen dieser Arbeit konstruiert

## 5. Benchmark der Optimierungsverfahren

schmalen Tal, umgeben von vielen lokalen Minima. Die Funktion ist definiert als

$$f_{\text{Miner-S-1}} \left\{ \begin{array}{l} 10 \cdot \sum_{i=1}^n \left( \eta_{1,2,50}(x_i) + \eta_{\frac{3}{2},4,100}(x_i) + \eta_{\frac{5}{2},6,2}(x_i) + \sigma(x_i) + \tau \right) \rightarrow \min \\ \eta_{p,a,c}(x_i) = -\frac{\exp\left(\frac{1}{c \cdot (x_i-p)^a + 1}\right) - 1}{e^{-\frac{\sqrt{2}}{a}}} \\ \text{mit } \sigma(x_i) = \frac{\sin^2((x_i-1) \cdot \pi) + \sin^3((x_i-1) \cdot \pi^2)}{\pi^3} \\ \tau = 1 + \frac{e^{\frac{4}{29}} - 1}{e^{-\frac{\sqrt{2}}{2}}} + \frac{e^{\frac{32}{61}} - 1}{e^{-\frac{\sqrt{2}}{6}}} \\ -1 \leq x_i \leq 6 \quad i = 1, 2, \dots, n \\ \mathbf{x} \in \mathbb{R}^n . \end{array} \right. \quad (5.6)$$

Das globale Optimum der Funktion befindet sich im Punkt  $\mathbf{x}^* \in \mathbb{R}^n$ ,  $\mathbf{x}^* = (1, \dots, 1)^T$  mit dem Funktionswert  $f(\mathbf{x}^*) = 0$ . Als Beispiel ist die 2-dimensionale MINER-S-1-Funktion<sup>19</sup> in Abbildung 5.22 dargestellt.

### 5.1.6.2. Zielstellung

Mit Hilfe der Testfunktion MINER-S-1 kann untersucht werden, wie gut der Optimierungsalgorithmus in der Lage ist, in einer stark multimodalen Umgebung das globale Minimum zu finden. Sie kann aber auch zum Test für robuste Optimierung eingesetzt werden, da mehrere lokale Optima deutlich robuster sind als das Globale. Mit dieser Testfunktion werden viele der Eigenschaften technischer Optimierungsaufgaben<sup>20</sup> abgebildet.

### 5.1.6.3. Test und Auswertung

In Abbildung 5.23 sind die Ergebnisse für die Testaufgabe mit 2 Variablen dargestellt. Sehr gut zu erkennen ist, dass mit dem NSGA-II und dem AHMOS das globale Minimum gefunden wird, wobei das hybride Verfahren deutlich besser ist. Zu erkennen ist aber auch, dass in beiden Verfahren häufig nur ein benachbartes Minimum gefunden wurde. Das NELDER-MEAD-Verfahren mit zufälligen Startpunkten ist für diese Aufgabe nicht effektiv, da es aufgrund der vielen lokalen Minima stets vor Erreichen des globalen Minimums abbricht.

In Abbildung 5.25 sind die Ergebnisse für die Testaufgabe mit 5 Variablen dargestellt. AHMOS kommt dem globalen Optimum im Mittel deutlich näher als die anderen Verfahren, erreicht es aber nicht exakt.

Im hybriden Verfahren AHMOS kommen beide Strategien „Globale Suche - Lokale Verfeinerung“ und die „Ersatzmodellssuche“ zum Einsatz. Die „Ersatzmodellssuche“ wirkt hierbei ebenfalls unterstützend für die Globale Suche (das Ersatzmodell bildet dabei die geglättete

<sup>19</sup>(5.6) mit  $n = 2$

<sup>20</sup>siehe Abschnitt 2.2



5.1. Test-Aufgaben für einkriterielle Optimierungsaufgaben

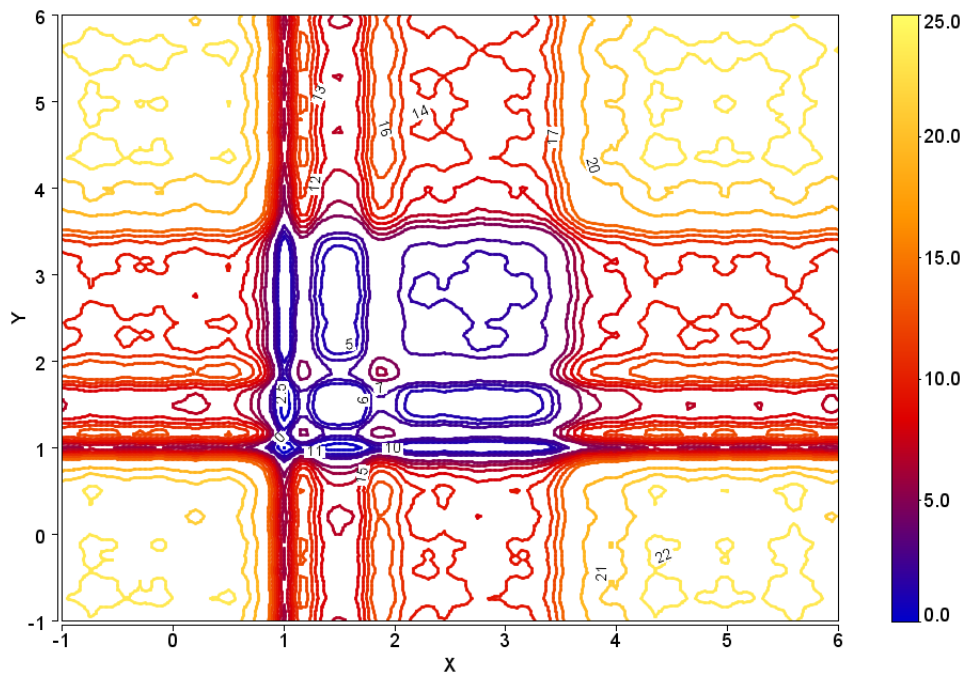
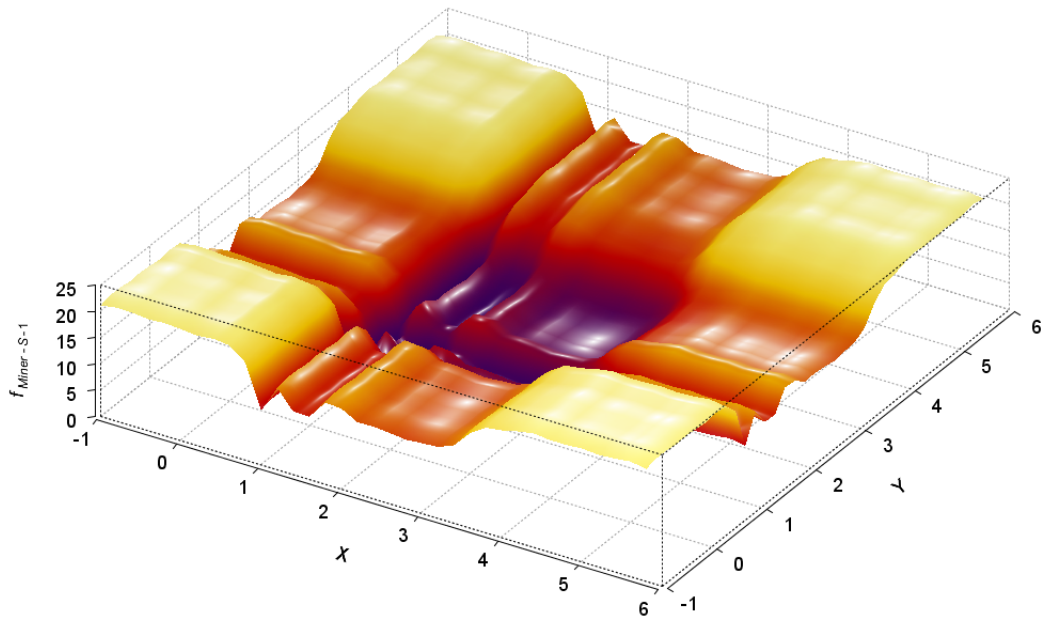


Abbildung 5.22.: Testfunktion: 2-dimensionale MINER-S-1

## 5. Benchmark der Optimierungsverfahren

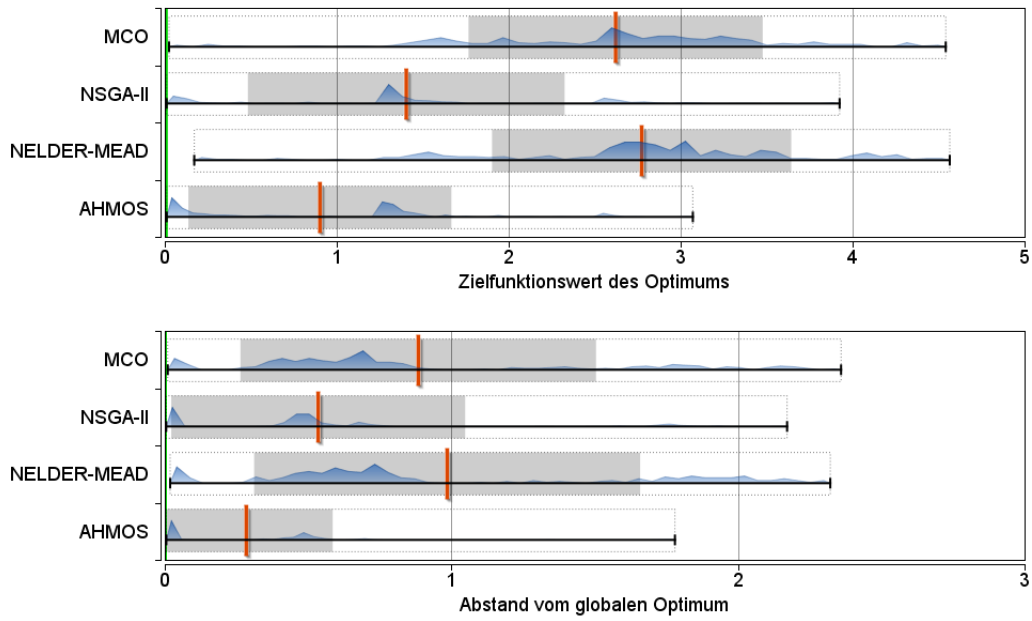


Abbildung 5.23.: Auswertung: MINER-S-1 mit 2 Variablen

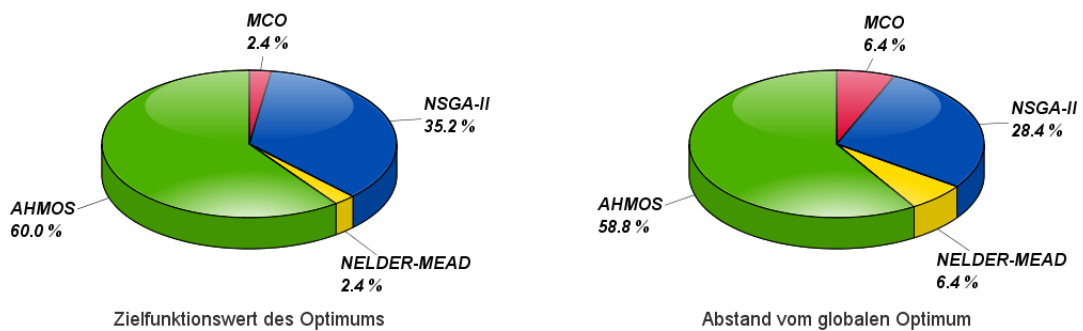


Abbildung 5.24.: Auswertung (bestes Verfahren): MINER-S-1 mit 2 Variablen

5.1. Test-Aufgaben für einkriterielle Optimierungsaufgaben



Abbildung 5.25.: Auswertung: MINER-S-1 mit 5 Variablen

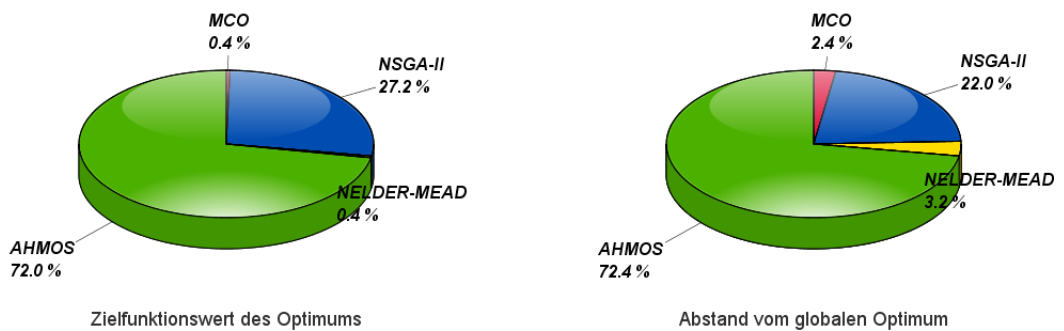


Abbildung 5.26.: Auswertung (bestes Verfahren): MINER-S-1 mit 5 Variablen

## 5. Benchmark der Optimierungsverfahren

Testfunktion ohne die vielen kleinen lokalen Minima ab). Die Strategie „Globale Suche - Lokale Verfeinerung“ kommt dann erfolgreich zum Einsatz, wenn die globale Optimierung und die „Ersatzmodellsuche“ bereits Lösungen in der Nähe der Minima gefunden haben.

In den Abbildungen 5.24 und 5.26 ist dargestellt, in welchem Verhältnis die verglichenen Verfahren die besten Lösungen ermittelt haben. Auch bei dieser Testaufgabe ist zu erkennen, dass besonders bei größerer Variablenanzahl das Hybridverfahren deutliche Vorteile aufweist.

**Ergebnis:** Die besten Resultate wurden jeweils mit dem hybriden Verfahren AHMOS erzielt.

### 5.2. Test-Aufgaben für multikriterielle Optimierungsaufgaben

Der Erfolg eines Optimierungsverfahrens bei der Lösung einer MOA lässt sich, im Gegensatz zu der Lösung einer EOA<sup>21</sup>, nicht daran messen, wie gut oder wie schnell ein globales Optimum gefunden wird. Da die Lösung der MOA eine Menge (die PARETO-Menge) möglicher Kompromisse ist, müssen bei der Bewertung die Ergebnismengen bewertet und verglichen werden. Dafür wurden eine Vielzahl möglicher Kriterien entwickelt (siehe u.a. [13, 14, 49, 58, 52]). Die folgenden Kriterien wurden zur Bewertung des Erfolges einer multikriteriellen Optimierung ausgewählt:

**Anzahl der Pareto-Punkte:** Dieses Kriterium gibt die durchschnittliche Anzahl der gefundenen Punkte der PARETO-Menge wieder. Ziel eines guten multikriteriellen Optimierungsverfahrens ist es, dass möglichst viele der gefundenen Lösungen zu dieser Menge gehören. Dieses Kriterium hat aber keine Aussagekraft darüber, wie dicht die ermittelte Approximation an der tatsächlichen PARETO-Menge liegen oder wie gut sie verteilt sind.

**Relative Dominanz<sup>22</sup>:** Ziel dieses Kriteriums ist es, eine Aussage zu treffen, welches der Verfahren besser zur PARETO-Menge konvergiert ist. Da meist die tatsächliche PARETO-Menge nicht bekannt ist, vergleicht man die Ergebnisse der einzelnen Optimierungsverfahren miteinander. Für diesen Relativvergleich werden zuerst die ermittelten PARETO-Mengen  $X_i$ <sup>23</sup> der zu vergleichenden Verfahren vereinigt

$$\check{M} = \bigcup_i X_i$$

und alle dominierten Individuen aus der Vereinigung  $\check{M}$  entfernt. Anschließend wird gezählt, wie viele Individuen der resultierenden PARETO-Menge  $\check{X}$  jeweils von den zu vergleichenden

---

<sup>21</sup>siehe Abschnitt 5.1

<sup>22</sup>siehe [58]

<sup>23</sup>mit  $i$  als Zähler für die zu vergleichenden Optimierungsverfahren

Optimierungsverfahren stammen

$$z^{(i)} = \sum_{\mathbf{x} \in X_i} \begin{cases} 1 & , \text{ wenn } \mathbf{x} \in \check{X} \\ 0 & , \text{ sonst} \end{cases} .$$

Jedem Verfahren wird der Wert

$$d_r^{(i)} = \frac{z^{(i)}}{|\check{X}|}$$

zugeordnet, der den relativen Anteil der vom Verfahren  $i$  ermittelten PARETO-Punkte an  $\check{X}$  darstellt.

**Konvergenzmetrik**<sup>24</sup>: Diese Metrik zur Bewertung von multikriteriellen Optimierungsproblemen wurde von KALYANMOY DEB vorgestellt<sup>25</sup>. Gegeben ist eine Menge  $X^*$  von PARETO-optimalen Lösungen des Optimierungsproblems. Diese Menge wird als Referenzmenge bezeichnet und dient als Basis für den Vergleich. Idealerweise ist  $X^*$  eine diskrete Teilmenge der PARETO-Menge. Da diese Menge in der Praxis häufig nicht bekannt ist, wird die PARETO-Menge  $\check{X}$  der Vereinigung der PARETO-optimalen Punkte aller Optimierungsergebnisse als Referenzmenge verwendet.

Die Konvergenzmetrik bewertet hauptsächlich die Konvergenz einer gegebenen Menge in Richtung der PARETO-Menge, die Verteilung der Punkte wird nicht betrachtet. Je kleiner der Wert der Konvergenzmetrik ist, desto näher liegt die Menge im Sinne dieser Metrik an der Referenzmenge und um so besser wird das Ergebnis der Optimierung bewertet.

Dazu wird der minimale normierte Abstand jedes Elements  $\mathbf{x}^{(j)} \in X_i, j = 1, \dots, |X_i|$  der PARETO-Menge  $X_i$ <sup>26</sup> der zu untersuchenden Verfahren von den Elementen  $\mathbf{x}^* \in X^*$  der Referenzmenge  $X^*$ <sup>27</sup>

$$\theta_i = \min_{\mathbf{x}^* \in X^*} \left( \sqrt{\sum_{m=1}^M \left( \frac{f_m(\mathbf{x}^{(i)}) - f_m(\mathbf{x}^*)}{f_m^{max} - f_m^{min}} \right)^2} \right)$$

berechnet. Die Symbole  $f_m^{max}$  und  $f_m^{min}$  stehen für die maximalen bzw. minimalen Funktionswerte der  $m$ -ten Zielfunktion in  $X^*$ . Die Konvergenzmetrik ist dann das arithmetische Mittel der normierten Abstände aller Elemente der untersuchten PARETO-Menge  $X_i$

$$CM(X_i) = \frac{1}{|X_i|} \sum_{i=1}^{|X_i|} \theta_i . \quad (5.7)$$

<sup>24</sup>siehe [58]

<sup>25</sup>siehe [13]

<sup>26</sup>mit  $i$  als Zähler für die zu vergleichenden Optimierungsverfahren

<sup>27</sup>in der Anwendung wird  $\check{X}$  verwendet

## 5. Benchmark der Optimierungsverfahren

Um die Vergleichbarkeit herzustellen, werden die Werte der Konvergenzmetrik auf das Intervall  $[0, 1]$  normiert.

Bei stochastischen Optimierungsverfahren ist es, ebenso wie bei den einkriteriellen OA, notwendig, die Optimierung mehrmals durchzuführen und so die mittlere Güte und die Streuung zu ermitteln. Um eine Vergleichbarkeit herzustellen, wird in den folgenden Tests für alle Optimierungsverfahren eine maximale Anzahl von 300 Funktionsauswertungen vorgegeben. Die Parametrisierung der einzelnen Verfahren ist in Anhang C aufgelistet. Jede Aufgabe wird 250 mal mit identischen Anfangsbedingungen optimiert und ausgewertet. Die Ergebnisse sind jeweils in Verteilungsplots<sup>28</sup> dargestellt. Weiterhin wurde für jeden der 250 Optimierungsläufe getestet, welches der Verfahren die beste Lösung<sup>29</sup> erzielt hat. Diese Ergebnisse werden anschaulich in einer Tortengrafik dargestellt.

Für die multikriteriellen Tests wurde das NELDER-MEAD-Verfahren nicht mit getestet, da es sich bei diesem Verfahren um ein einkriterielles Verfahren handelt.

### 5.2.1. Testfunktion: DTLZ1

#### 5.2.1.1. Definition

Die DTLZ1-Testfunktion<sup>30</sup> für  $M$  Zielfunktionen ist definiert als

$$f_{DTLZ1} \begin{cases} \frac{1}{2}x_1x_2 \dots x_{M-1} (1 + g(\mathbf{x})) \rightarrow \min \\ \frac{1}{2}x_1x_2 \dots (1 - x_{M-1}) (1 + g(\mathbf{x})) \rightarrow \min \\ \vdots \\ \frac{1}{2}x_1 (1 - x_2) (1 + g(\mathbf{x})) \rightarrow \min \\ \frac{1}{2} (1 - x_1) (1 + g(\mathbf{x})) \rightarrow \min \\ \text{mit } g(\mathbf{x}) = 100 \cdot \left( |\mathbf{x}| + \sum_{i=1}^{|\mathbf{x}|} \left( x_i - \frac{1}{2} \right)^2 - \cos \left( 20\pi \left( x_i - \frac{1}{2} \right) \right) \right) \\ 0 \leq x_i \leq 1 \quad i = 1, 2, \dots, M + k - 1 \\ \mathbf{x} \in \mathbb{R}^{M+k-1} \end{cases} \quad (5.8)$$

und besitzt  $M+k-1$  Variablen  $\mathbf{x} = (x_1, \dots, x_{M+k-1})^T$ . Empfohlen wird  $k = 5$ . Die PARETO-Menge ist direkt angebar. Die PARETO-optimalen Lösungen liegen auf der Hyperebene

$$\sum_{m=1}^M f_m = \frac{1}{2}.$$

<sup>28</sup>Erläuterung des Verteilungsplots: siehe Anhang B

<sup>29</sup>hinsichtlich der Anzahl der gefundenen PARETO-optimalen Lösungen, der Relativen Dominanz und der Konvergenzmetrik

<sup>30</sup>vorgestellt von DEB, THIELE, LAUMANN und ZITZLER, siehe [13, 14]

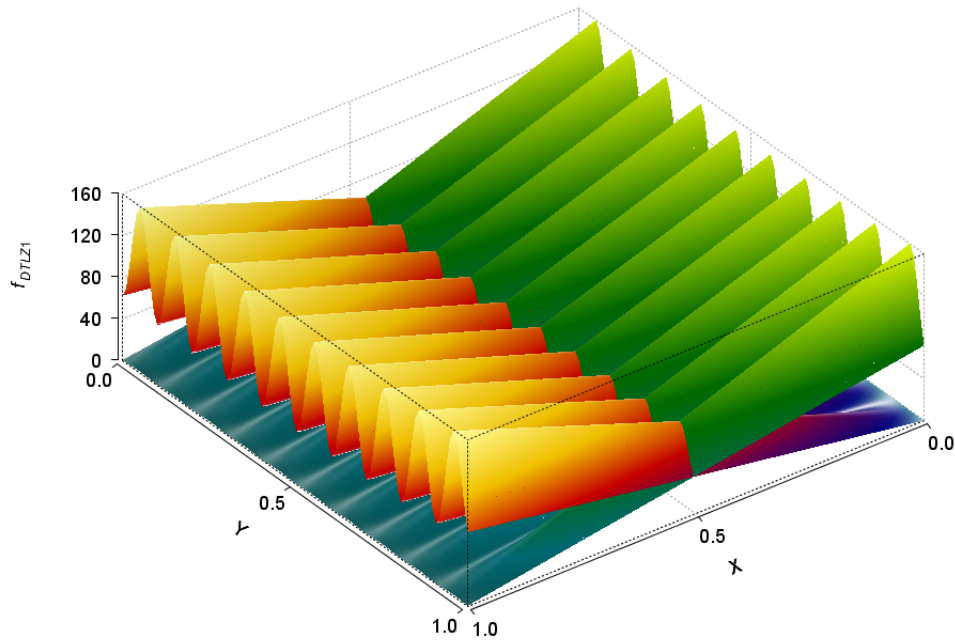


Abbildung 5.27.: Testfunktion: DTLZ1 mit 2 Zielen und 2 Variablen

In dieser Arbeit wird die DTLZ1-Testfunktion mit  $M = 2$  Zielen und 6 Variablen eingesetzt ( $k = 5$ ). Damit ergibt sich aus (5.8):

$$f_{DTLZ1} \begin{cases} \frac{1}{2}(1 - x_1)(1 + g(\mathbf{x})) \rightarrow \min \\ \frac{1}{2}x_1(1 + g(\mathbf{x})) \rightarrow \min \\ \text{mit } g(\mathbf{x}) = 100 \cdot \left(6 + \sum_{i=1}^6 (x_i - \frac{1}{2})^2 - \cos(20\pi(x_i - \frac{1}{2}))\right) \\ 0 \leq x_i \leq 1 \quad i = 1, 2, \dots, 6 \\ \mathbf{x} \in \mathbb{R}^6. \end{cases} \quad (5.9)$$

Als Beispiel ist die DTLZ1-Funktion<sup>31</sup> in Abbildung 5.27 dargestellt.

### 5.2.1.2. Zielstellung

Die Herausforderung bei dieser Testfunktion besteht darin, dass die PARETO-Front zu der Hyperebene konvergiert. Der Suchraum enthält sehr viele lokale PARETO-Fronten<sup>32</sup>, die von der globalen Front ablenken.

<sup>31</sup>(5.8) mit  $k = 1$  und  $M = 2$

<sup>32</sup> $(11^k - 1)$

## 5. Benchmark der Optimierungsverfahren

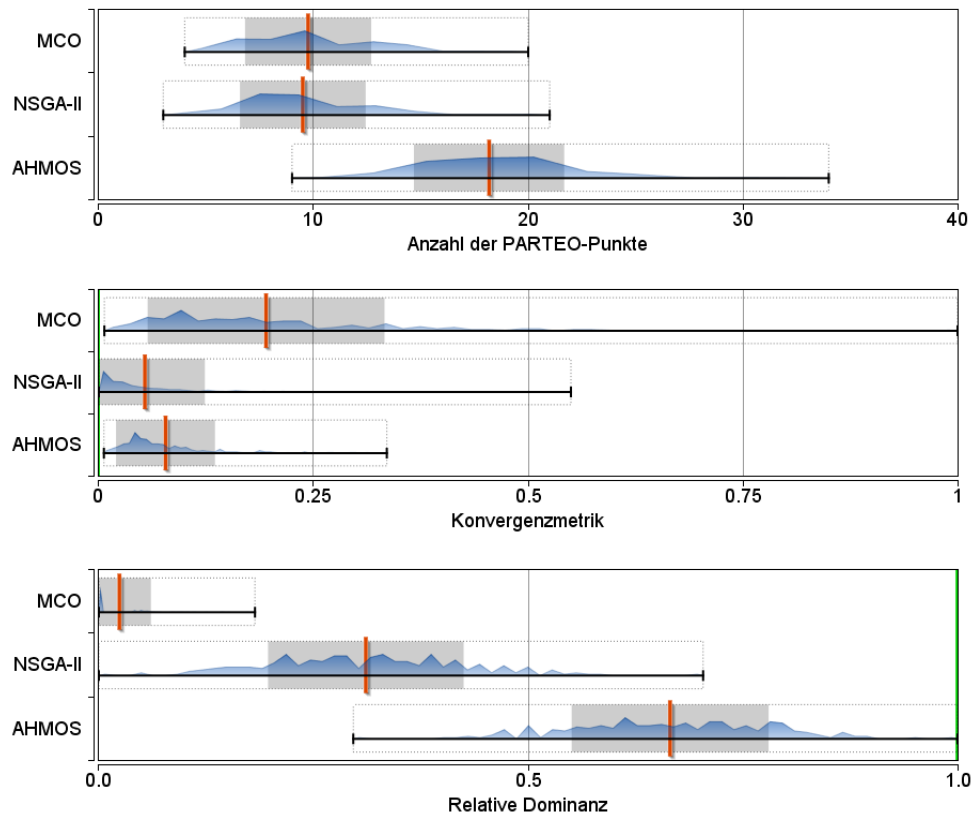


Abbildung 5.28.: Auswertung: DTLZ1 mit 2 Zielen und 6 Variablen

### 5.2.1.3. Test und Auswertung

In Abbildung 5.28 sind die Ergebnisse für die Testaufgabe mit 2 Zielen und 6 Variablen (Formel (5.9)) dargestellt. Der AHMOS findet im Mittel die meisten PARETO-Punkte und zeigt in der *Relativen Dominanz* deutliche Vorteile. In der *Konvergenzmetrik* wird er schlechter bewertet als der NSGA-II. Daraus lässt sich schließen, dass mit dem hybriden Verfahren die Verteilung der Punkte in der PARETO-Front verbessert wurde, jedoch der NSGA-II in einigen Bereichen der PARETO-Front besser konvergiert ist, dafür aber andere Bereiche entweder noch nicht oder nicht gut untersucht hat.

In Abbildung 5.29 ist dargestellt, in welchem Verhältnis die verglichenen Verfahren die besten Lösungen ermittelt haben. Auch hier ist zu erkennen, dass die hybriden Strategien für diese Testaufgabe hinsichtlich der Anzahl der PARETO-Punkte und der *Relativen Dominanz* deutliche Vorteile aufweisen, hinsichtlich der *Konvergenzmetrik* aber schlechter bewertet werden.

**Ergebnis:** Mit dem hybriden Verfahren AHMOS wurden im Mittel die besten Ergebnisse erzielt, jedoch sind mit dem NSGA-II einige Bereiche der PARETO-Front besser konvergiert.



## 5.2. Test-Aufgaben für multikriterielle Optimierungsaufgaben

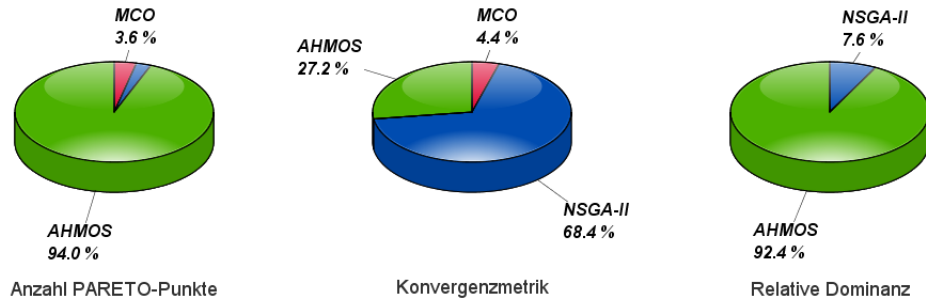


Abbildung 5.29.: Auswertung (bestes Verfahren): DTLZ1 (2 Ziele und 6 Variablen)

### 5.2.2. Testfunktion: DTLZ2

#### 5.2.2.1. Definition

Diese Testfunktion wird als skalierbare multikriterielle Testfunktion in [14] eingeführt. Die PARETO-Menge ist ein Ausschnitt der Einheitssphäre und ihre Elemente müssen der Bedingung

$$\sum_{m=1}^M (f_m(\mathbf{x})) = 1$$

genügen. Damit ist die PARETO-Menge nicht-konvex, was manchen Verfahren Probleme bereitet.

Die DTLZ2-Testfunktion für  $M$  Zielfunktionen ist definiert als

$$f_{DTLZ2} \left\{ \begin{array}{l} (1 + g(\mathbf{x})) \cos\left(\frac{\pi}{2}x_1\right) \cos\left(\frac{\pi}{2}x_2\right) \cdots \cos\left(\frac{\pi}{2}x_{M-2}\right) \cos\left(\frac{\pi}{2}x_{M-1}\right) \rightarrow \min \\ (1 + g(\mathbf{x})) \cos\left(\frac{\pi}{2}x_1\right) \cos\left(\frac{\pi}{2}x_2\right) \cdots \cos\left(\frac{\pi}{2}x_{M-2}\right) \sin\left(\frac{\pi}{2}x_{M-1}\right) \rightarrow \min \\ (1 + g(\mathbf{x})) \cos\left(\frac{\pi}{2}x_1\right) \cos\left(\frac{\pi}{2}x_2\right) \cdots \sin\left(\frac{\pi}{2}x_{M-2}\right) \rightarrow \min \\ \vdots \\ (1 + g(\mathbf{x})) \cos\left(\frac{\pi}{2}x_1\right) \sin\left(\frac{\pi}{2}x_2\right) \rightarrow \min \\ (1 + g(\mathbf{x})) \sin\left(\frac{\pi}{2}x_1\right) \rightarrow \min \\ \text{mit } g(\mathbf{x}) = \sum_{i=1}^{M+k-1} \left(x_i - \frac{1}{2}\right)^2 \\ 0 \leq x_i \leq 1 \quad i = 1, 2, \dots, M+k-1 \\ \mathbf{x} \in \mathbb{R}^{M+k-1} \end{array} \right. \quad (5.10)$$

und besitzt  $M+k-1$  Variablen  $\mathbf{x} = (x_1, \dots, x_{M+k-1})^T$ . Für die Untersuchungen wurden die Testfunktionen mit

## 5. Benchmark der Optimierungsverfahren

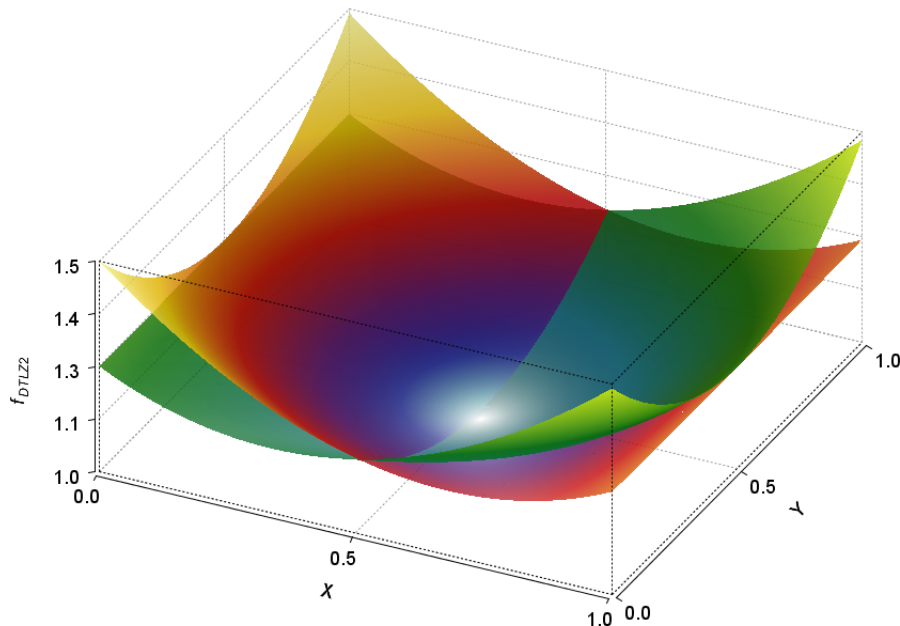


Abbildung 5.30.: Testfunktion: DTLZ2 mit 2 Zielen und 2 Variablen

- $M = 2$  und  $k = 4$
- $M = 4$  und  $k = 4$

gewählt. In Abbildung 5.30 ist zur Veranschaulichung die DTLZ2 mit 2 Variablen und 2 Zielen visualisiert.

### 5.2.2.2. Zielstellung

Mit der DTLZ2 wird untersucht, wie gut die Optimierungsverfahren in der Lage sind, zu der nichtkonvexen PARETO-Front zu konvergieren. Weiterhin wird untersucht, wie gut die Verfahren mit vielen Zielfunktionen umgehen können.

### 5.2.2.3. Test und Auswertung

In Abbildung 5.31 sind die Ergebnisse für die Testaufgabe mit 2 Zielen und 5 Variablen dargestellt. Bei dieser Aufgabe findet der NSGA-II im Mittel die meisten PARETO-Punkte. Das hybride Verfahren AHMOS zeigt hingegen wiederum in der *Relativen Dominanz* und in der *Konvergenzmetrik* deutliche Vorteile. Das bedeutet, dass mit dem hybriden Verfahren mehr Punkte ermittelt werden, die nahe an der PARETO-Front liegen, auch wenn der NSGA-II insgesamt mehr PARETO-Punkte findet. Die in Abbildung 5.33 dargestellten Ergebnisse für die Testaufgabe mit 4 Zielen und 7 Variablen zeigen die gleichen Eigenschaften.

## 5.2. Test-Aufgaben für multikriterielle Optimierungsaufgaben

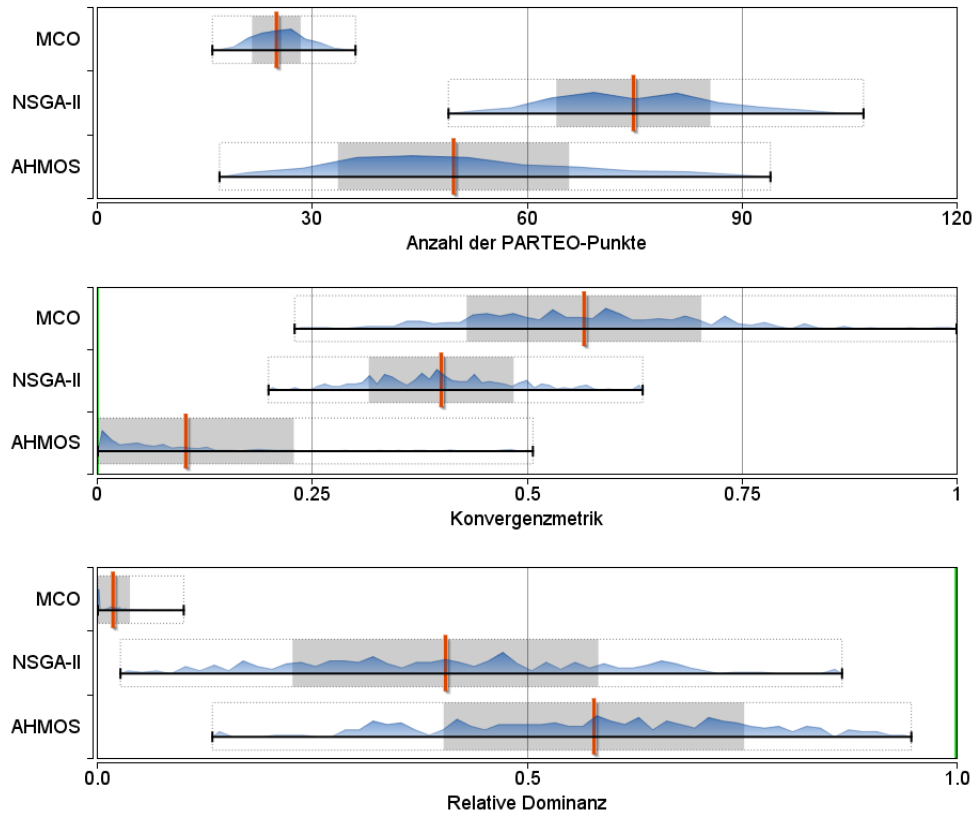


Abbildung 5.31.: Auswertung: DTLZ2 mit 2 Zielen und 5 Variablen

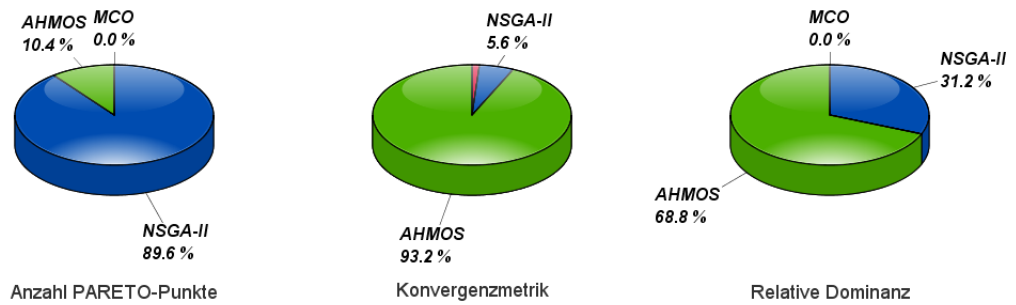


Abbildung 5.32.: Auswertung (bestes Verfahren): DTLZ2 (2 Ziele und 5 Variablen)

## 5. Benchmark der Optimierungsverfahren

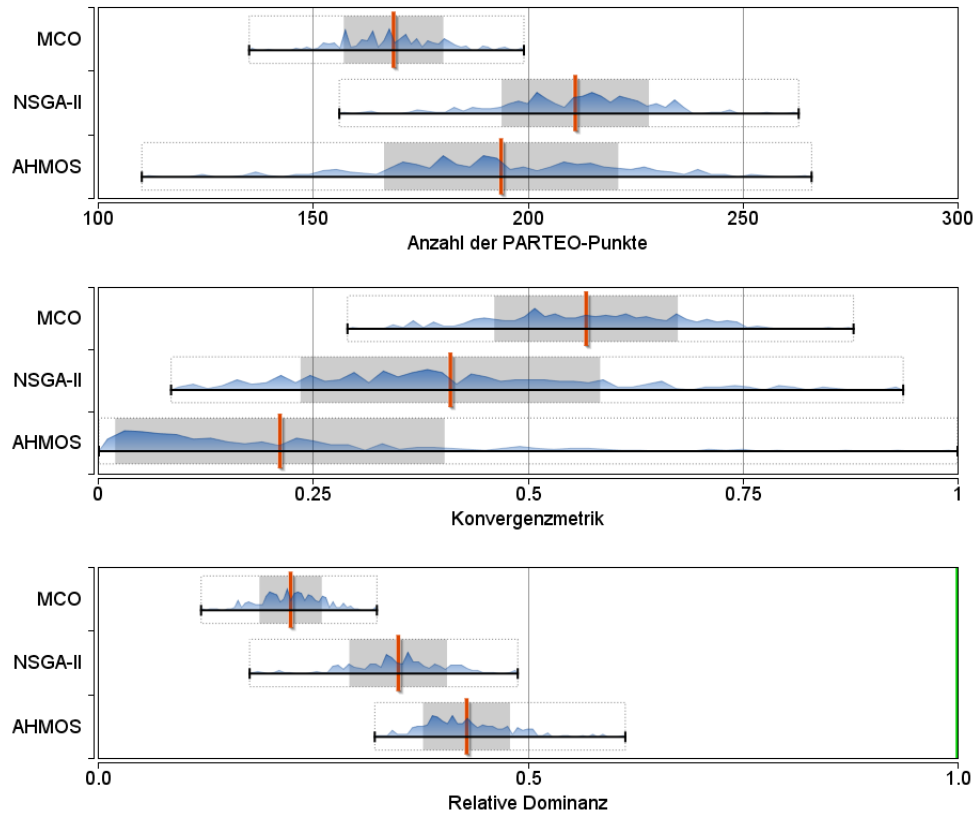


Abbildung 5.33.: Auswertung: DTLZ2 mit 4 Zielen und 7 Variablen

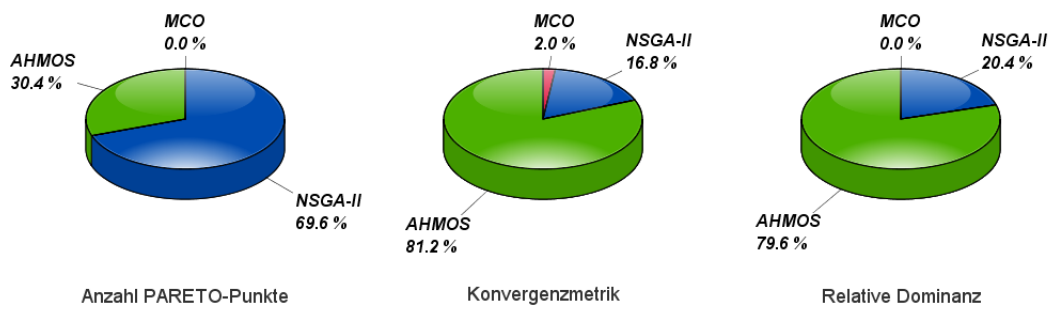


Abbildung 5.34.: Auswertung (bestes Verfahren): DTLZ2 (4 Ziele und 7 Variablen)

## 5.2. Test-Aufgaben für multikriterielle Optimierungsaufgaben

In den Abbildungen 5.32 und 5.34 ist dargestellt, in welchem Verhältnis die verglichenen Verfahren die besten Lösungen ermittelt haben. Auch hier ist zu erkennen, dass mit den hybriden Strategien für diese Testaufgabe hinsichtlich der Anzahl der PARETO-Punkte vom NSGA-II am besten gelöst wurde, hinsichtlich der *Relativen Dominanz* und der *Konvergenzmetrik* zeigt der AHMOS deutliche Vorteile.

**Ergebnis:** Mit dem hybriden Verfahren AHMOS wurden im Mittel die besten Ergebnisse erzielt, wobei aber der NSGA-II meist mehr PARETO-Punkte gefunden hat.

### 5.2.3. Testfunktion: Miner-M-1

#### 5.2.3.1. Definition

Die Testfunktion MINER-M-1<sup>33</sup> ist charakterisiert durch eine nichtzusammenhängende und nichtkonvexe PARETO-Menge mit nichtlinearen Nebenbedingungen mit vielen lokalen PARETO-Fronten. Durch die Restriktionen ist die zulässige Menge mit vielen Löchern versehen. Eine Visualisierung der Restriktionen ist in Abbildung 5.36 dargestellt. Die Testfunktion ist definiert als

$$f_{Miner-M-1} \left\{ \begin{array}{l} 2 \cdot \left( \frac{x_1}{x_2} \right) + \frac{\cos(\pi x_1 \sum_{i=1}^M x_i)}{10} \rightarrow \min \\ 2^2 \cdot \left( \frac{x_2}{x_3} \right) + \frac{\cos(\pi x_2 \sum_{i=1}^M x_i)}{10} \rightarrow \min \\ \vdots \\ 2^M \cdot \left( \frac{x_M}{x_1} \right) + \frac{\cos(\pi x_M \sum_{i=1}^M x_i)}{10} \rightarrow \min \\ \prod_{i=1}^M \sin(30x_i) < \frac{1}{8} \\ \prod_{i=1}^M \sin(20x_i) < \frac{1}{2} \\ 1 \leq x_i \leq 2 \quad i = 1, 2, \dots, M \\ \mathbf{x} \in \mathbb{R}^M. \end{array} \right. \quad (5.11)$$

In Abbildung 5.35 sind die 2 Zielfunktionen der Testfunktion mit 2 Variablen visualisiert<sup>34</sup>. Für die Untersuchung wurde die MINER-M-1 mit  $M = 2$  und  $M = 5$  verwendet.

#### 5.2.3.2. Zielstellung

Das Ziel des Tests mit der Testfunktion MINER-M-1 ist es, Funktionen mit beschränkten zulässigen Bereichen und nichtzusammenhängender, nichtkonvexer PARETO-Front abzubilden. Weiterhin wird getestet, wie die Verfahren in diesem Kontext mit einer höheren Anzahl von

<sup>33</sup>diese Testfunktion wurde im Rahmen dieser Arbeit definiert

<sup>34</sup>ohne Nebenbedingungen

## 5. Benchmark der Optimierungsverfahren

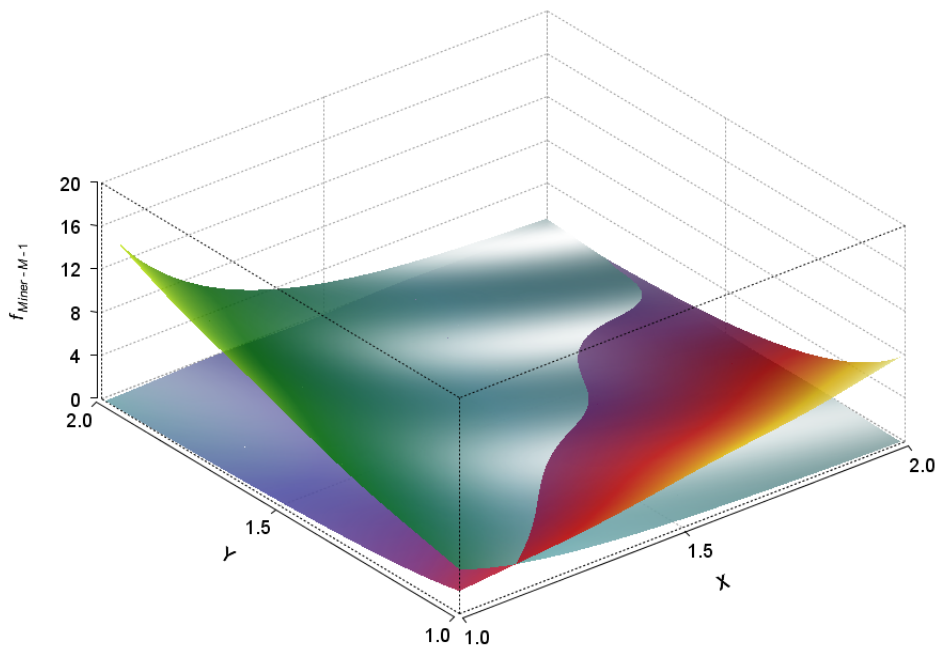


Abbildung 5.35.: Testfunktion: MINER-M-1 mit 2 Zielen und 2 Variablen, ohne Restriktionsbereiche (siehe Abbildung 5.36)

Zielfunktionen umgehen können. Die Form der Nebenbedingungen (viele „Löcher“) dient weiterhin zur Simulation von Optimierungsaufgaben, bei denen viele unlösbare Varianten innerhalb der zulässigen Parameterintervalle entstehen, wie es bei technischen Optimierungsaufgaben vorkommt.

### 5.2.3.3. Test und Auswertung

In den Abbildungen 5.37 und 5.39 sind die Ergebnisse für die Testaufgabe dargestellt. Auch bei dieser Aufgabe findet der NSGA-II im Mittel die meisten PARETO-Punkte. Das hybride Verfahren AHMOS zeigt hingegen bei der *Relativen Dominanz* und bei der *Konvergenzmetrik* deutliche Vorteile. Auffällig ist allerdings die breite Streuung der Werte. Dies liegt darin begründet, dass durch die niedrige Anzahl an Funktionsauswertungen und der Restriktionen zum Teil nur wenige zulässige Punkte gefunden wurden und damit manche Bereich der nichtzusammenhängenden PARETO-Front nicht erreicht werden.

In Abbildung 5.38 und 5.40 wird dargestellt, in welchem Verhältnis die verglichenen Verfahren die besten Lösungen ermittelt haben. Der NSGA-II findet zwar in den meisten Fällen mehr PARETO-Punkte, jedoch sind die vom AHMOS gefundenen Punkte deutlich besser.

**Ergebnis:** Mit dem hybriden Verfahren AHMOS werden die besten Ergebnisse erzielt.

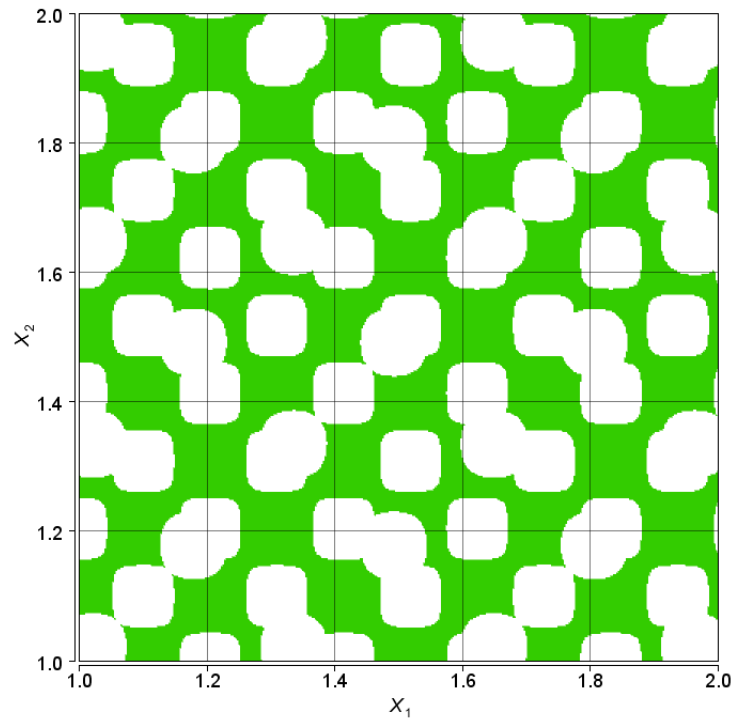


Abbildung 5.36.: Restriktionen der Testfunktion: MINER-M-1 mit 2 Zielen und 2 Variablen, grün: zulässiger Bereich

### 5.3. Zusammenfassung und Schlussfolgerung

Der Einsatz der hybriden Strategien, die in Kapitel 4 vorgestellt und im hybriden Algorithmus AHMOS umgesetzt wurden, bietet einen deutlichen Nutzen. Für alle der untersuchten Testaufgaben konnte gezeigt werden, dass eine Verbesserung gegenüber den Ausgangsverfahren<sup>35</sup> durch den Einsatz der hybriden Methoden möglich ist.

#### Einkriterielle Aufgaben

Für alle einkriteriellen Testaufgaben konnte gezeigt werden, dass bei einer fest vorgegebenen Anzahl von Funktionsauswertungen durch den Einsatz der hybriden Strategien die Güte der gefundenen Optima deutlich verbessert werden konnte im Vergleich zu den Ausgangsverfahren. Es ist somit empfehlenswert, das Verfahren AHMOS für EOA zu verwenden.

#### Multikriterielle Aufgaben

Für die multikriteriellen Testaufgaben konnte ebenfalls gezeigt werden, dass bei einer fest vorgegebenen Anzahl von Funktionsauswertungen durch den Einsatz der hybriden Strategien die

<sup>35</sup>NSGA-II und NELDER-MEAD-Simplex-Verfahren

## 5. Benchmark der Optimierungsverfahren

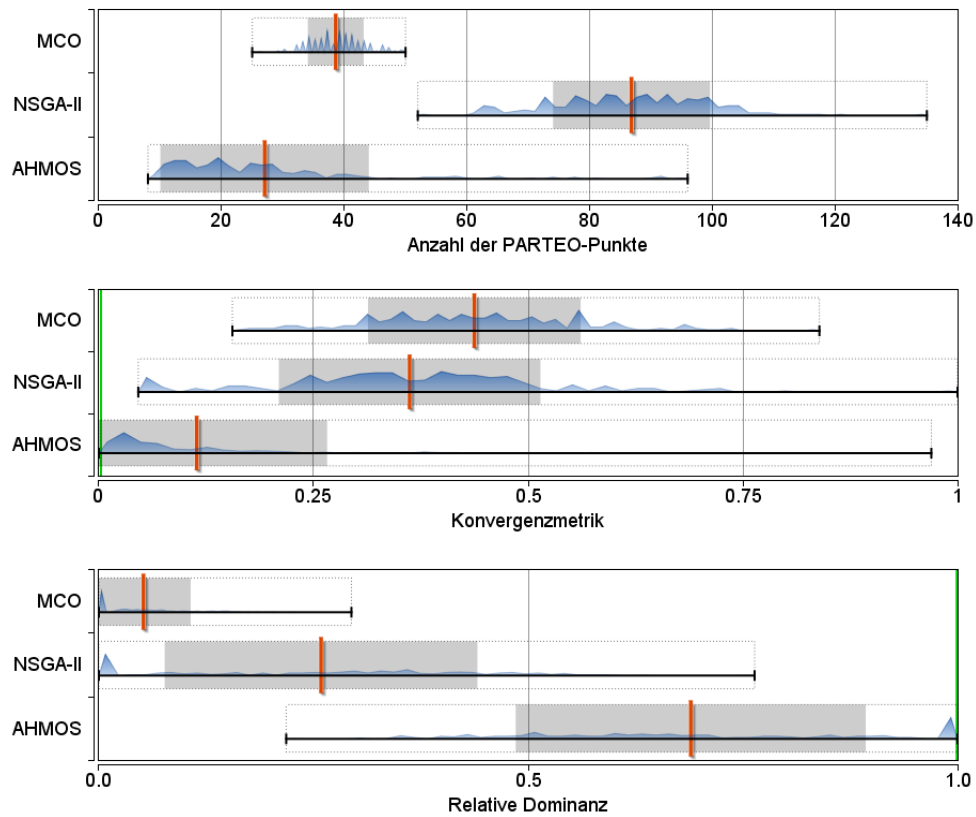


Abbildung 5.37.: Auswertung: MINER-M-1 mit 2 Zielen und 2 Variablen

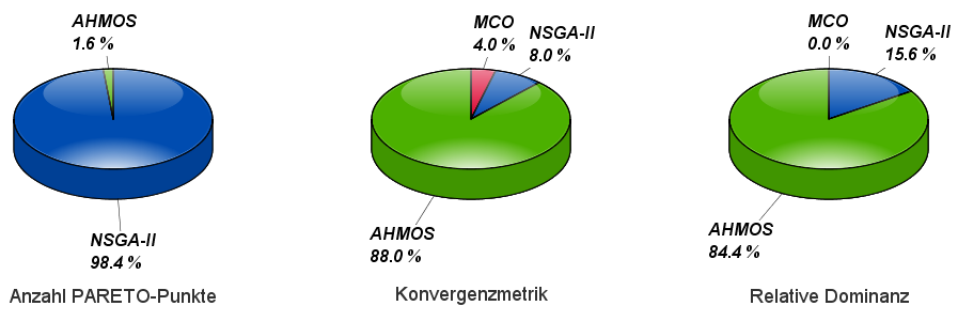


Abbildung 5.38.: Auswertung (bestes Verfahren): MINER-M-1 (2 Ziele und 2 Variablen)



### 5.3. Zusammenfassung und Schlussfolgerung

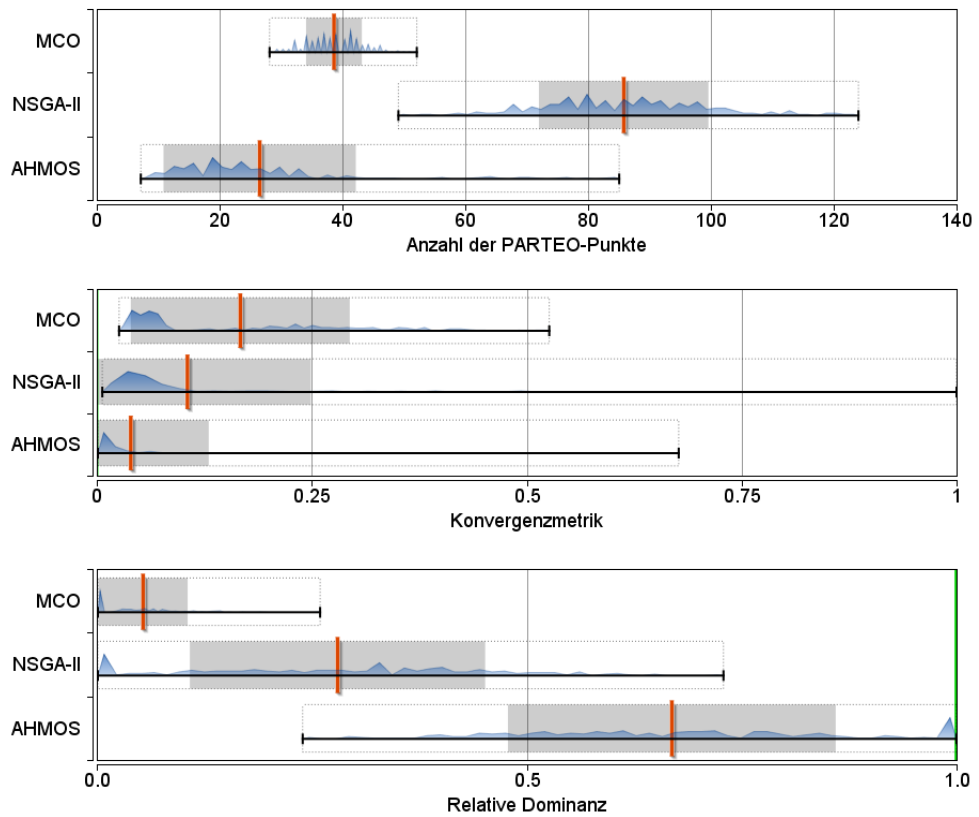


Abbildung 5.39.: Auswertung: MINER-M-1 mit 5 Zielen und 5 Variablen

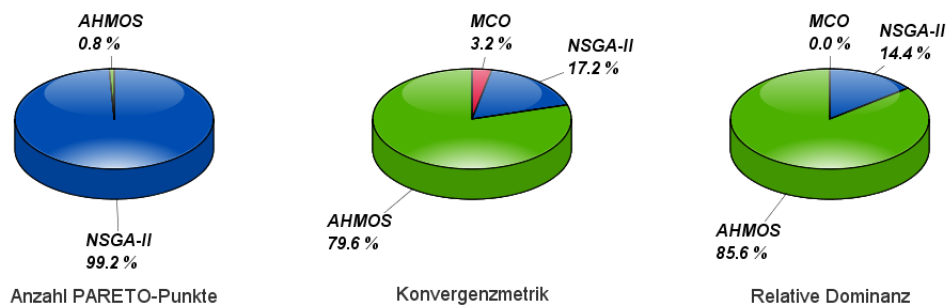


Abbildung 5.40.: Auswertung (bestes Verfahren): MINER-M-1 (5 Ziele und 5 Variablen)

## 5. Benchmark der Optimierungsverfahren

Güte der gefundenen Optima verbessert werden konnte. Dabei zeigte sich jedoch, dass für die einzelnen Testaufgaben nicht alle Kriterien verbessert wurden. Bei der Testfunktion DTLZ1 hat der AHMOS die meisten PARETO-Punkte gefunden und überragt in der *Relativen Dominanz*, zeigt aber Nachteile in der *Konvergenzmetrik*. Bei den anderen Testfunktionen zeigen die Kriterien für die Konvergenz<sup>36</sup> sehr deutliche Vorteile für den AHMOS, allerdings ist die Anzahl der gefundenen PARETO-Punkte kleiner als beim NSGA-II. Es wurden somit weniger PARETO-optimale Lösungen ermittelt, diese waren aber besser als bei den Vergleichsverfahren. Trotz dieser Einschränkungen ist es empfehlenswert, das Verfahren AHMOS für MOA zu verwenden.

---

<sup>36</sup>Relative Dominanz und Konvergenzmetrik

# 6

## Kapitel 6.

---

# Zusammenfassung und Ausblick

In dieser Arbeit wurden Optimierungsverfahren untersucht, die dafür geeignet sind, komplexe technische Aufgabenstellungen zu lösen. Optimierungsaufgaben zur Lösung derartiger Problemstellungen sind besonders dadurch gekennzeichnet, dass die Auswertung der Zielfunktionen oft sehr aufwändig sein kann und aus diesem Grund in der Praxis meist nur eine sehr begrenzte Anzahl von Funktionswertberechnungen ausgeführt werden können. Im ersten Teil der Arbeit wurden deterministische und stochastische Optimierungsverfahren eingeführt, die für diese Art von Problemen eingesetzt werden können.

Im Mittelpunkt der Arbeit standen *hybride Strategien*, die durch Kombination verschiedener Optimierungsmethoden das Ziel verfolgen, die Effektivität bei der Lösung derartiger Aufgaben zu erhöhen. Dabei wurden die hybriden Methoden sowohl für ein- als auch für multikriterielle Optimierungsaufgaben betrachtet.

Die erste untersuchte Methode ist „*Globale Suche - lokale Verfeinerung*“ (Abschnitt 4.2). Bei dieser Methode wird mit einem globalen Optimierungsverfahren begonnen, den gesamten zulässigen Parameterraum zu durchsuchen, um die Gebiete zu identifizieren, in denen sich potentiell optimale Lösungen befinden. Ist ein Bereich identifiziert, wird mit einem lokal arbeitenden Verfahren fortgesetzt, um schnell zu dem Optimum zu konvergieren.

Für die globale Suche wird der reellwertige Genetische Algorithmus *NSGA-II*<sup>1</sup> und für die lokale Verfeinerung wird das *NELDER-MEAD-Simplex-Verfahren*<sup>2</sup> eingesetzt.

Als zweite Methode wurde die „*Ersatzmodellsuche*“ erarbeitet (Abschnitt 4.3.1.2). Bei dieser Methode handelt es sich um eine Strategie, die von den *adaptiven Ersatzmodellverfahren*<sup>3</sup> abgeleitet wurde. Dazu wird aus den verfügbaren Informationen<sup>4</sup> ein Approximationsmodell erzeugt. Bildet dieses die Zielfunktionen und Nebenbedingungen der originalen Optimierungs-

---

<sup>1</sup>siehe Abschnitt 3.3.3.1

<sup>2</sup>siehe Abschnitt 3.2.2.1

<sup>3</sup>siehe Abschnitt 3.3.4

<sup>4</sup>Zwischenergebnisse der globalen Suche

## 6. Zusammenfassung und Ausblick

aufgabe mit ausreichender Genauigkeit ab, wird die Optimierung temporär auf dem Ersatzmodell fortgesetzt. Die Ergebnisse (das gefundene Optimum bei EOA bzw. einige Elemente der PARETO-Menge bei MOA) werden mit den Originalzielfunktionen nachgerechnet und in die Population der globalen Suche eingefügt. Die „Ersatzmodellssuche“ kann innerhalb einer Optimierung oder im Anschluss an eine Optimierung als Nachiteration angewendet werden.

Als Verfahren zur Erzeugung des Approximationsmodells kommt das *Moving-Least-Square-Verfahren*<sup>5</sup> zum Einsatz.

Das hybride Optimierungsverfahren AHMOS (Abschnitt 4.5), in dem diese beiden Methoden ihre Anwendung finden, ist zur Lösung komplexer technischer Aufgabenstellungen gut geeignet. Es koppelt effektiv die positiven Eigenschaften von unterschiedlichen Verfahren:

1. Globale Suche im gesamten Parameterraum,
2. Lokale Optimierung mit guter Konvergenz in den Bereichen, die vielversprechend sind,
3. Ausnutzung eines Ersatzmodells während der Optimierung und als Nachiteration.

In umfangreichen Tests (Kapitel 5) wurden die herausgearbeiteten Vorteile der hybriden Strategien, die im Verfahren AHMOS implementiert wurden, anhand von geeigneten Testfunktionen nachgewiesen. Die von den gewählten Testfunktionen getesteten Eigenschaften sind typisch für technische Aufgabenstellungen, so dass AHMOS auch für diese Aufgaben effektiv einsetzbar sein wird.

## Ausblick

Das Verfahren AHMOS stellt weiterhin einen leicht erweiterbaren Baukasten für hybride Optimierungsmethoden dar, so dass weitere Strategien leicht integriert werden können. Die in Abschnitt 4.5.3 empfohlenen Methoden und Ansätze, z.B.

- Robuste Optimierung
- Adaptive Verfahrensparameteranpassung
- Insel-Modelle
- andere Verfahren als globale und lokale Optimierer

stellen erfolgversprechende Weiterentwicklungsmöglichkeiten dar.

---

<sup>5</sup>siehe Abschnitt 4.3.1.2

# Index

- Abkühlung, 41
- Ableitung, 18
- Abstand, 57, 93
- adaptiv, 42
- Adaptive Sharing, 34
- AHMOS, 61, 67, 103, 108
- Algorithmus
  - Ameisen-, 31
  - Bergsteiger, 30
  - Bienen-, 31
  - Genetischer, 31
  - Hillclimber, 30
  - Metropolis, 41
  - Partikelschwarm, 31
  - Simulated Annealing, 31
- Alpen-Problem, 13, 84
- Ameisen, 38
- analytisch, 1
- Ant Systems, 39
- Approximation, 53, 56, 57
- Approximationsmodell, 42, 46, 58, 107
- Approximationsschema, 57
- Approximationsverfahren, 56
- Asynchron Hybrid, 47
- Aufwand, 29, 43
- automatisieren, 1
  
- Basisfunktion, 56
- Basisfunktionen, 56
- Baukasten, 108
- Bauteil, 1
  
- Benchmark, 67
- Berg, 84
- Beste Lösung, 1, 8
- Bestimmtheitsmaß, 58, 62
- Biologie, 31
- Blackbox-Probleme, 3, 10
- Breitesuche, 46
  
- CFD, 3
- Clusteranalyse, 55
- CPU, 4, 14
- Crowding-Distance, 33
  
- Definitionsbereich, 11
- Dichtefunktion, 41
- differenzierbar, 17
- Dimension, 10
- disjunkte Teilbereiche, 46
- disjunkte Teilmengen, 13
- diskret, 11
- dominiert, 8
- Dynamic Crowding, 33
  
- Effektivität, 42
- Einflussbereich, 57
- Einflussfaktor, 37
- Einheitssphäre, 97
- Einschränkung, 55
- Energiezustand, 40
- Entscheidungskriterien, 48
- Entwicklungsprozess, 1, 10
- Ersatzaufgabe, 59

## Index

- Ersatzmodell, 42, 44, 55, 58–60, 62, 65, 108
- Ersatzmodell-Verfahren, 42
- Ersatzmodellsuche, 55, 58, 62, 65
- Ersatzoptimierungsaufgabe, 23, 53
- Ersatzzielfunktion, 55, 65
- Erstarrungsprozess, 40
- Evolutionsstrategien, 32
- Expansion, 19
- Extrema
  - lokale, 13
- FEM, 3
- Fitness, 32–34, 53
- Fitnessbewertung, 37
- Fortschrittsanalyse, 52, 53
- FOURIER-Ordnung, 11
- Funktion
  - ALPENTAL, 84
  - DEB2, 81
  - DTLZ1, 94
  - DTLZ2, 97
  - HIMMELBLAU, 78
  - Miner-M-1, 101
  - MINER-S-1, 87
  - nichtlinear, 1
  - RASTRIGIN, 68
  - ROSENBROCK, 74
- Funktionen
  - nichtlinear, 1
- Funktionsauswertung, 2, 4, 46
- Funktionssystem, 1
- Funktionswertberechnung
  - aufwändige, 1
- Funktionswertbestimmung, 10
- GAUSSfunktion, 57
- GAUSSsches Normalgleichungssystem, 57
- Generation, 32, 62
- Genetik, 31, 32
- Genetische Algorithmen, 32
- Geschwindigkeit, 37
- Geschwindigkeitsupdate, 36, 38
- Getriebeabstimmung, 3
- Gewicht, 24
- Gewichtsfunktion, 57
- Gitterenergie, 40
- Gleichungssystem, 57, 58
  - gleichverteilt, 28
  - global, 2, 47
- Globale Suche, 47, 48, 64, 108
- Gradient, 2, 3, 11, 17
- Heuristik, 28, 31, 32, 45, 49
- Hierarchischer Hybrid, 47
  - homogen, 40
- Homogenität, 49, 53
- Hybrid, 45, 56
  - hybride Strategien, 107
  - hybriden Strategien, 103
- Hyper-Intervalle, 23
- Hyperebene, 95
- Individuum, 32
- Initialisierung, 30, 42
- Insel-Modelle, 65
- Inselmodell, 46
- Intelligenz, 35
- Interpolation, 56
- Intervall, 22
- Intervallteilung, 22
- Iteration, 32
- Iterationsschritt, 22, 43
- Kanalgeometrie, 3
- Kettenrad, 3
- Kettentriebslayout, 3
- Koeffizienten, 56, 57
- Koeffizientenfunktion, 57
- Kombination, 45
- kombinatorische Probleme, 40
- Kompromiss, 1, 8, 11

- konkav, 25
- Konstante
  - BOLTZMANN, 41
  - LIPSCHITZ, 21
- Kontraktion, 19
- Konvergenz, 4, 36, 41, 54, 56, 58, 93, 108
- Konvergenzgeschwindigkeit, 35
- Konvergenzkriterien, 62
- Konvergenzmetrik, 93
- konvex, 17, 24, 25
- konvexe Gebiete, 48
- Kreuzung, 32, 47
- Kriging, 42
- Kristallgitter, 40
- Kurbeltriebsgeometrie, 3
- Kurbelwelle, 3, 11
  
- Lösungsraum, 17
- Latin-Hypercube-Methode, 30
- Latin-Hypercube-Sampling, 30
- LEGENDRE-Polynome, 56, 121
- Linearkombination, 32
- LIPSCHITZ
  - Konstante, 22
  - Optimierung, 21
  - stetig, 21
- Lockstoffe, 38
- lokal, 2
- Lokale Suche, 49, 65, 108
- Lokale Verfeinerung, 47
  
- Materialmodell, 3
- Maximum, 123
- MENDELSche Gesetze, 31, 32
- Mersenne-Twister, 29
- Methode
  - BENSON-, 26
  - Beschränkungs-, 25
  - Cross-Entropy-, 42
  - Gewichtete-Metrik-, 26
  - Gewichtete-Summen-, 24
  - klassische, 23
  - Monte-Carlo-, 29
  - Nutzenfunktions-, 26
  - PARETO-Scan, 27, 55
  - $\varepsilon$ -Constraint-, 25
- Methode der kleinsten Quadrate, 51
- Minimum, 123
- Mittelwert, 50, 123
- MKS, 3
- Modell, 46
  - parametrisch, 1
  - statistisches, 28
- Modellabgleich, 3, 11
- Modellerstellung, 65
- Modellfehler, 55
- Modellqualität, 62
- Modelltraining, 42
- Modellunsicherheit, 55
- Monte-Carlo-Methode, 29
- Monte-Carlo-Optimierung, 29
- MrBOA, 35
- MSOPS, 33
- multikriteriell, 1, 11, 23, 27, 30, 33, 37, 40, 41, 43, 53, 92
- multimodal, 68, 87
- Multiple Regression, 56
- Mutation, 32, 33, 47
  
- Nachbarn, 35
- Nachbarschaft, 30, 41, 43
- Nachiteration, 44, 60, 65, 108
- Nebenbedingung, 2, 10, 11, 17, 25, 81
- Nebenbedingungen, 7
- Netzwerk, 4
- nicht zusammenhängend, 11
- nichtdominiert, 8
- nichtlinear, 7
- nichtzusammenhängend, 101
- No-Free-Lunch, 2

## Index

- Normierung, 24, 94
- NSGA-II, 33, 61, 62
- Nutzenfunktion, 26
  
- Operator, 32
  - Geschwindigkeitsupdate, 36
  - Hybrid, 47
  - Kreuzung, 32
  - Mutation, 33
- Optimierung, 1, 2, 10, 44, 60
- Optimierungsaufgabe, 7
- Optimierungsvariablen, 7
- Optimierungsverfahren, 2
- Optimum, 62
  - global, 2, 13, 22, 28, 41, 47, 48, 69
  - lokal, 2, 10, 13, 17, 22, 37, 43, 47, 69
- orthogonales Funktionensystem, 121
  
- Parallelisierbarkeit, 4
- Parallelisierung, 4, 10, 14, 64
- Parameterraum, 1
- PARETO
  - Front, 8, 12, 26, 34, 44, 47, 53, 65
  - Menge, 8, 12, 23, 25, 38, 47, 53, 55, 58, 59, 62, 92
  - optimal, 8
- PARETO
  - Scan, 27, 55
- Partikel, 35, 36
- Periode, 29
- Pheromon
  - Update, 39
- Pheromone, 38
- Physik, 31
- Pipeline Hybrid, 47
- Plateau, 87
- Polynommodell, 51
- Population, 32, 35, 49, 53, 62
  - aktuelle, 62
- Populationsgröße, 52
  
- Problem-Relaxation, 46
- Pseudozufallszahlen, 15, 28
  
- quadratisches Mittel, 56
  
- Rand, 17
- Rang, 33
- Ranking, 33
- Rechencluster, 4
- Rechenkern, 4, 14
- Referenz-Verfahren, 29
- Reflektion, 19
- Regressionsanalyse, 51
- Regressionsmodell, 51
- Relative Dominanz, 92
- Relativvergleich, 53, 92
- Restriktion, 1, 10
- Restriktionsbereiche
  - nichtzusammenhängend, 13
  - schmal, 13
- Richtung, 17
- Richtungsänderung, 36
- Robuste Optimierung, 46, 65
- Robustheit, 65
- RODRIGUES-Formel, 121
- Rundreiseproblem, 40
  
- Schmelze, 40
- Schranken, 25
- Schrumpfen, 19
- Schwarm, 36
- Schwarmoptimierung, 35
- Schwarmverhalten, 35
- Selektion, 32
- Shared Population, 47
- Sharing Intensity, 34
- signifikante Verbesserung, 28
- Simplex, 18, 65
- Simulated Annealing, 46
- Simulation, 1
- Simulationssoftware, 2



- SMS-EMOA, 33
- soziale Abläufe, 45
- SPEA2, 33
- Stützstelle, 57
- Stagnation, 52, 54
- Standardabweichung, 50
- Startpopulation, 32
- Startwert, 44
- Statistik, 45
- statistische Vergleiche, 123
- statistisches Modell, 28
- steilster Anstieg, 17
- Stochastic Design Approvement, 30
- Streuung, 123
- Suchraum, 2, 10, 46, 55, 95
- Suchstrategie, 49
- Symbiose, 49
  
- Tabuliste, 40
- Tal, 74, 84, 87
- technische Systeme, 1
- Teilintervall, 22
- Temperatur, 40
- Tensorprodukt, 56
- Test-Aufgaben, 68
- Tests, 67, 108
- Tiefensuche, 46
- Tschebyscheff-Norm, 26
  
- Umschalten, 48
- Umschaltkriterien, 49, 53
- Unabhängigkeit, 29
- Uniformität, 29
- unlösbare Varianten, 1, 2, 12, 52
- Unsicherheit, 43
  
- Varianten
  - unlösbare, 2
- Variation, 32
- Variationskoeffizient, 50
- Vektoroptimierung, 7
  
- Ventiltrieb, 3
- Verbesserung, 10, 53
- Vererbung, 32
- Verfahren
  - adaptive Ersatzmodell-, 14, 42
  - Ameisen, 38
  - Ameisen-, 31
  - Ameisenkolonie, 38
  - Branch-And-Bound, 45
  - CG-, 17
  - des steilsten Abstiegs, 17
  - deterministische, 17
  - DiRect, 21, 62
  - Downhill-Simplex-, 18
  - Entscheidungsbaum-, 45
  - Evolutionäre, 32
  - exakte, 17
  - Genetische Algorithmen, 32
  - Gradienten-, 17, 49
  - gradientenfrei, 18
  - HOUSEHOLDER, 57
  - kombinierte, 45
  - Metropolis, 41
  - Moving-Least-Squares-, 56, 57
  - naturinspiriert, 14, 31, 45
  - NELDER-MEAD, 49, 69
  - NELDER-MEAD, 18, 61
  - Nelder-Mead-, 62
  - NEWTON-, 17
  - NSGA-II, 61, 69
  - Partikelschwarm, 31, 35
  - SHUBERT, 21
  - Simulated Annealing, 40
  - SQP, 17
- Verfahrensklasse, 15
- Verfahrensparameteranpassung, 65
- Verhaltensweisen, 31
- Verschlechterung, 41, 69
- Verteilung, 93

## *Index*

BOLTZMANN, 41  
Verteilungsplot, 69, 123  
virtuelle Entwicklung, 1  
  
Wahrscheinlichkeit, 41  
Wahrscheinlichkeitsmodell, 35, 40  
Wahrscheinlichkeitsverteilung, 35  
Wertebereich, 29, 55  
  
zeitaufwändig, 1  
Ziele  
    widersprüchlich, 1, 11  
Zielfunktion, 1, 2, 7, 17, 18, 25, 55  
Zielfunktionsraum, 42  
Zielraum, 47  
Zielwert, 123  
Zufallszahl, 36  
Zufallszahlen, 28  
    Generatoren, 28  
    Pseudo-, 28  
zulässige Menge, 7, 10, 26

## Literaturverzeichnis

- [1] ABRAMOWITZ, MÍLTON; STEGUN, IRENE A.: Handbook of Mathematical Functions. National Bureau of Standards, Washington, D.C., Applied Mathematics Series 55, 1964.
- [2] AFZERI, NUKHAIE IBRAHIM: Hybrid Optimization of Pin type fixture Configuration for Free Form Workpiece. International Journal of Science Engineering and Technology, Vol. 2, No. 1, 2008.
- [3] AHN, CHANG W.: Advances in Evolutionary Algorithms. Axel Springer-Verlag, 2006.
- [4] ALT, WALTER: Nichtlineare Optimierung. Friedr. Vieweg & Sohn Verlagsgesellschaft, Braunschweig/Wiesbaden, 2002.
- [5] BALLUCHI, A.; BICCHI, A.; PADRONI, G.; ROSSI, C.; SANGIOVANNI-VINCENTELLI, A.: Hybrid Optimization Problems in Automotive Applications. In Proceedings of IFAC International Workshop on Motion Control, John Wiley & Sons, Seiten 273-278, 1998.
- [6] BALSACANTO, EVA; PEIFER, MARTIN; BANGA, JULIO R.; TIMMER, JENS; FLECK, CHRISTIAN: Hybrid optimization method with general switching strategy for parameter estimation. BMC Systems Biology Journal, BioMed Central Ltd., 2008.
- [7] BLUM, CHRISTIAN; PUCHINGER, JACOB; RAIDL, GÜNTHER; ROLI, ANDREA: A Brief Survey on Hybrid Metaheuristics. In: B. Filipic and J. Silc, editors, Proceedings of BIO-MA 2010 - 4th International Conference on Bioinspired Optimization Methods and their Applications, Jozef Stefan Institute, Ljubljana, Slovenia, ISBN: 978-961-264-017-0, 2010.
- [8] BYATT, DAVID: Convergent Variants of the Nelder-Mead-Algorithm. Masterarbeit, University of Canterbury, 2000.
- [9] CORNE, DAVID; DORIGO, MARCO; GLOVER, FRED; DASGUPTA, DIPANKAR; MOSCATO, PABLO; POLI, RICCARDO; PRICE, KENNETH V. : New Ideas in Optimization. McGraw-Hill, Maidenhead, UK, 1999.
- [10] DEB, KALYANMOY: Exercices to Multi-Objective Optimization Using Evolutionary Algorithms. Ammendement für die erste Ausgabe von: Multi-Objective Optimization using Evolutionary Algorithms, John Wiley & Sons, Chichester, New York, Weinheim, Brisbane, Toronto, Singapore, 2001.

## Literaturverzeichnis

- [11] DEB, KALYANMOY: Kanpur Genetic Algorithms Laboratory - GA Resources, Constraint Handling Techniques. Internetquelle (<http://www.iitk.ac.in/kangal/resources.shtml#4> am 19.September 2011), Indian Institute of Technology Kanpur, 2005.
- [12] DEB, KALYANMOY: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Chichester, New York, Weinheim, Brisbane, Toronto, Singapore, 2001.
- [13] DEB, KALYANMOY; JAIN, SACHIN: Running Performance Metrics for Evolutionary Multi-Objective Optimization. KanGAL Report No. 2002004, Seiten 16-20, Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology Kanpur, India, 2002.
- [14] DEB, KALYANMOY; THIELE, LOTHAR; LAUMANN, MARCO; ZITZLER, ECKART: Scalable Test Problems for Evolutionary Multi-Objective Optimization. TIK-Technical Report No. 112, Institut für Technische Informatik und Kommunikationsnetze, ETH Zürich, 2001.
- [15] DE BOER, P.T.; KROESE, D.; MANNOR, S.; RUBINSTEIN, R.: A Tutorial on the Cross-Entropy Method. Annals of Operations Research, Vol. 134, Seiten 19-67, 2005.
- [16] DORIGO, MARCO: Optimization, Learning and Natural Algorithms. Dip. Elettronica, Politecnico di Milano, PhD thesis, 1992.
- [17] ENGELN-MULLGES, G.; NIEDERDRENK, K. UND WODICKA, R.: Numerik- Algorithmen. 9. Auflage, Springer-Verlag, Berlin Heidelberg, 2005.
- [18] DOMSCHKE, WOLFGANG; HÄSELBARTH, L. ; SCHOLL, ARMIN: WISU-Lexikon Operations Research. Lange Verlag Düsseldorf, 2003.
- [19] FINKEL, DANIEL E.: DIRECT Optimization Algorithm User Guide. Technical Report (<http://www.ncsu.edu/crsc/reports/ftp/pdf/crsc-tr03-11.pdf> am 5.Mai 2011), Center for Research in Scientific Computation, North Carolina State University, 2003.
- [20] HACKER, KURT A.; EDDY, JOHN; LEWIS, KEMPER E.: Efficient Global Optimization using Hybrid Genetic Algorithms. Journal Image Rochester NY, (September), Seiten 1-11, 2002.
- [21] HACKER, KURT A.; EDDY, JOHN; LEWIS, KEMPER E.: Tuning a Hybrid Optimization Algorithm by Determining the Modality of the Design Space. In Proceedings of the Asme Design Engineering Technical Conference, Vol. 2, Seiten 773-782, 2001.
- [22] HEDAR, ABDEL-RAHMAN: Global Optimization Test Problems. Internetquelle ([http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar\\_files/TestGO.htm](http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO.htm) am 14.September 2011), 2005.
- [23] HINZE, THOMAS: Prinzipien zur Erzeugung von Zufallszahlen in der Informatik. Technische Universität Dresden, Fachvortrag im Rahmen des Promotionsverfahrens am 23.09.2002, 2002.

- [24] JACKSON, DUNHAM: Fourier Series and Orthogonal Polynomials. Mathematical Assoc. of America, Carus Mathematical Monographs No. 6, 1941.
- [25] KENNEDY JAMES; EBERHART RUSSEL: A new optimizer using particle swarm theory. In Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Seiten 25-34, 1995.
- [26] KENNEDY, JAMES; EBERHART, RUSSEL: Particle swarm optimization. In Proceedings of the Fourth IEEE International Conference on Neural Networks in Perth (ISBN: 0-7803-2768-3), Journal Neural Networks, Vol. 6, Seiten 1942-1948, 1995.
- [27] KIRKPATRICK, S.; GELLAT, C.D.; VECCHI, M.P.: Optimization by Simulated Annealing. Science, Vol. 220, Seiten 671-680, 1983.
- [28] KNOWLES, JOSHUA: ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Transactions on Evolutionary Computation 10, Seiten 50-66, 2006.
- [29] KNOWLES, JOSHUA; HUGES, E. J.: Multi Objective Optimization on a Budget of 250 Evaluations. Lecture notes on computer science. In Proceedings of Third International Conference on Evolutionary Multi-Criterion Optimization, Vol. 3410, Seiten 176-190, 2005.
- [30] KUNLE, MATTHIAS: Entwicklung und Untersuchung von Moving Least Square Verfahren zur numerischen Simulation hydrodynamischer Gleichungen. Dissertation, Eberhard-Karls-Universität Tübingen, 2001.
- [31] KUTNER, MICHAEL H.: Applied Linear Statistical Models. The McGraw-Hill/Irwin series operations and decision sciences, McGraw-Hill Irwin Verlag, 2005.
- [32] KUX, STEFFEN; KUNERT, GERD: Online-Dokumentation der IAV-EngineeringToolbox. Version 4.11, IAV GmbH, 2011.
- [33] KUX, STEFFEN: Schulungsunterlagen Multikriterielle Optimierung mit der IAV-EngineeringToolbox. Version 4.11, IAV GmbH, 2011.
- [34] KUX, STEFFEN; PARSCHE UWE: Effektive Optimierungsverfahren im Auslegungsprozess für Kettentriebe. MTZ 01/2009, Jahrgang 70, Seiten 58-65, Springer Automotive Media, Springer Fachmedien Wiesbaden GmbH, 2009.
- [35] LAMMERT, DANIEL: Direct Search Algorithms. Internetquelle (<http://www.scai.fraunhofer.de/fileadmin/ArbeitsgruppeTrottenberg/WS0809/seminar/Lammert.pdf> am 02.Juni 2011), 2008.

## Literaturverzeichnis

- [36] LEVIN, DAVID: The Approximation Power of Moving Least-Squares. *Mathematics of Computation*, Volume 67, Nummer 224, Seiten 1517-1531, 1998.
- [37] MARSAGLIA, GEORG: Monkey Tests for Random Number Generators. *Computer & Mathematics with Applications*, Vol. 9, Seiten 1-10, 1993.
- [38] MATSUMOTO, MAKOTO; NISHIMURA, TAKUJI: Mersenne twister. A 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Transactions on Modeling and Computer Simulation*, Vol. 8, Seiten 3-30, 1998.
- [39] METROPOLIS, NICHOLAS; ROSENBLUTH A.; ROSENBLUTH, MARSHALL; TELLER, A.; TELLER, EDWARD: Equation of state calculation by fast computing machines. In *Journal of Chemical Physics*, Band 21, Seiten 1087-1092, 1953.
- [40] MÜHLENBEIN, H.; SCHOMISCH, D.; BORN, J.: The Parallel Genetic Algorithm as Function Optimizer. *Parallel Computing*, Vol. 17, Seiten 619-632, 1991.
- [41] NAGAR, A.; HERAGU, S. S.; HADDOCK, J.: A meta-heuristic algorithm for a bicriteria scheduling problem. *Annals of Operations Research*, 1995.
- [42] OBERLE, HANS JOACHIM: Optimierung. Vorlesungsskript, Sommersemester 2009, Universität Hamburg, 2009.
- [43] PELIKAN, MARTIN: Hierarchical Bayesian Optimization Algorithm - Towards a New Generation of Evolutionary Algorithms. *Studies in Fuzziness and Soft Computing*, Vol. 170, Springer Berlin Heidelberg, 2005.
- [44] PRESS, WILLIAM H.; TEUKOLSKY, SAUL A.; VETTERLING, WILLIAM T.; FLANNERY, BRIAN P.: *Numerical Recipes in C: The Art of Scientific Computing*. Second Edition, Cambridge University Press, 1992.
- [45] PUCHINGER, JAKOB; RAIDL, GÜNTHER: Combining Metaheuristics and Exact Algorithms in Combinatorial Optimization: A Survey and Classification. In *Proceedings of the First International Work-Conference on the Interplay Between Natural and Artificial Computation*, Part II, Vol. 3562 of LNCS, Seiten 41-53, Springer, 2005.
- [46] PYL, PAUL THEODOR; RUDOLPH, KONRAD LUDWIG MORITZ: Ameisenalgorithmus zur Lösung komplexer Optimierungsprobleme. Internetquelle ([http://www.activevb.de/tutorials/tut\\_antalgo/tut\\_antalgo.html](http://www.activevb.de/tutorials/tut_antalgo/tut_antalgo.html) am 02.Juni 2011), 2008.
- [47] RAIDL, GÜNTHER; CHWATAL, ANDREAS: Heuristische Optimierungsverfahren. Vorlesungsskript, Wintersemester 2010/2011, Technische Universität Wien, 2011.

- [48] RITTER, HELGE: Entropie und Information. Vorlesungsskript, persönliche Information, Universität Bielefeld, 2005.
- [49] RÖBER, MARCEL: Multikriterielle Optimierungsverfahren für rechenzeitintensive technische Aufgabenstellungen. Diplomarbeit, TU Chemnitz/IAV GmbH, 2010.
- [50] ROSENBROCK, H. H.: An automatic method for finding the greatest or least value of a function. The Computer Journal 3, Seiten 175–184, doi:10.1093/comjnl/3.3.175, ISSN 0010-4620, MR0136042, 1960.
- [51] RUDOLPH, GÜNTHER: Praktische Optimierung. Vorlesungsskript, Sommersemester 2011, Technische Universität Dortmund, 2011.
- [52] SCHMIDT, HANSJÖRG: Parallelisierung Ersatzmodell-gestützter Optimierungsverfahren. Diplomarbeit, Technische Universität Chemnitz/IAV GmbH, 2009.
- [53] SCHNABEL, DANNY: Verfahren der ein- und multikriteriellen Optimierung für Anwendungen in der Motorenentwicklung am Beispiel der Steuertriebsauslegung. Diplomarbeit, Hochschule Mittweida - University of Applied Sciences/IAV GmbH, 2005.
- [54] SCHULZE, HENDRIK: Entwicklung, Untersuchung und Implementierung von Parallelen Evolutionären Algorithmen für die Modellpartitionierungskomponente parallelMAP. Diplomarbeit, Universität Leipzig, 1998.
- [55] SCHUSTER, JULIA: Untersuchung von Optimierungsverfahren der ein- und multikriteriellen Optimierung mit diskreten Parametern für Anwendungen in der Antriebsstrangauslegung. Bachelorarbeit, Hochschule Mittweida - University of Applied Sciences/IAV GmbH, 2010.
- [56] SIMPSON, T. W.; MAUERY, T. M.; KORTE, J.J.; AND MISTREE, F.: Comparison of Response Surface and Kriging Models for Multidisciplinary Design Optimization. In Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis & Optimization, St. Louis, MI, AIAA-98-4755, 1998.
- [57] STAMER, HEIKO: Ant Algorithmen für kombinatorische Optimierungsprobleme. Skript zu Problemseminar „Artificial Life“, Institut für Informatik, Universität Leipzig, Internetquelle (<http://www.gaos.org/~stamer/AAfkO.pdf> am 02.Juni 2011), 2001/2005.
- [58] STÖCKER, MARTIN: Untersuchung von Optimierungsverfahren für rechenzeitaufwändige technische Anwendungen in der Motorenentwicklung. Diplomarbeit, TU Chemnitz/IAV GmbH, 2007.
- [59] THÄMMIG, STEFAN: Untersuchung von Approximationsverfahren und Verfahren zur Validierung der Modellgüte von Ersatzmodellen. Diplomarbeit, Hochschule Mittweida - University of Applied Sciences/IAV GmbH, 2010.

## *Literaturverzeichnis*

- [60] TÖRN, A.; ZILINSKAS, A.: Global Optimization. Lecture Notes in Computer Science, Nummer 350, Springer-Verlag, Berlin, 1989.
- [61] WILL, JOHANNES; MOST, THOMAS: Metamodell of optimized Prognosis (MoP) - an Automatic Approach for User Friendly Parameter Optimization. In Proceedings of the WOST 6.0 (Weimarer Optimierung- und Stochastik Tage 6.0), Weimar, dynardo GmbH, 2009.
- [62] WOLPERT, DAVID H.; MACREADY, WILLIAM G.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, Vol. 1, Nummer 1, Seiten 67–82, 1997.
- [63] YEN, JON; LIAO, JAMES C.; LEE, BOGJU; RANDOLPH, DAVID: A Hybrid Approach to Modelling Metabolic Systems Using Genetic Algorithm and Simplex Method. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, Vol. 28, Issue 2, Seiten 173-191, Digital Object Identifier: 10.1109/3477.662758, 1998.
- [64] YUN, YOUNGSU: Study on Adaptive Hybrid Genetic Algorithm and Its Applications to Engineering Design Problems. Thesis (Eng. D.), Waseda University, 2005.
- [65] ZIMMERMANN, UWE T.: Konvexe und Diskrete Optimierung. Vorlesungsskript, Sommersemester 2008, Technische Universität Braunschweig, (<http://imo.rz.tu-bs.de/~degbers/konvdisk08> am 05.Oktober 2011), 2008.



# A

## Anhang A.

---

# Legendre-Polynome

Die LEGENDRE-Polynome<sup>1</sup> (siehe [24, 1]) bilden auf dem Intervall  $[-1, 1]$  ein orthogonales Funktionensystem<sup>2</sup>. Sie sind die partikulären Lösungen der LEGENDRE'schen Differentialgleichung und spielen eine wichtige Rolle in der theoretischen Physik.

Als orthogonales Funktionensystem sind sie auch gut als Basisfunktionen für die Regression geeignet.

Als Bildungsvorschrift dient die RODRIGUES-Formel:

$$P_n(x) = \frac{1}{2^n n!} \cdot \frac{d^n}{dx^n} (x^2 - 1)^n.$$

Die ersten LEGENDRE-Polynome für  $n = 0, \dots, 6$  lauten:

$$\begin{aligned} P_0(x) &= 1 \\ P_1(x) &= x \\ P_2(x) &= \frac{1}{2}(3x^2 - 1) \\ P_3(x) &= \frac{1}{2}(5x^3 - 3x) \\ P_4(x) &= \frac{1}{8}(35x^4 - 30x^2 + 3) \\ P_5(x) &= \frac{1}{8}(63x^5 - 70x^3 + 15x) \\ P_6(x) &= \frac{1}{16}(231x^6 - 315x^4 + 105x^2 - 5) \\ &\vdots \end{aligned}$$

---

<sup>1</sup>benannt nach dem Mathematiker ADRIEN-MARIE LEGENDRE

<sup>2</sup>Beweis der Orthogonalität siehe [24]



# B

## Erläuterung: Verteilungsplot

Der Verteilungsplot (siehe Abbildung B.1) wird im Kapitel 5 eingesetzt, um statistische Vergleiche von Ergebnisgrößen der Optimierungsverfahren übersichtlich und vergleichbar zu visualisieren.

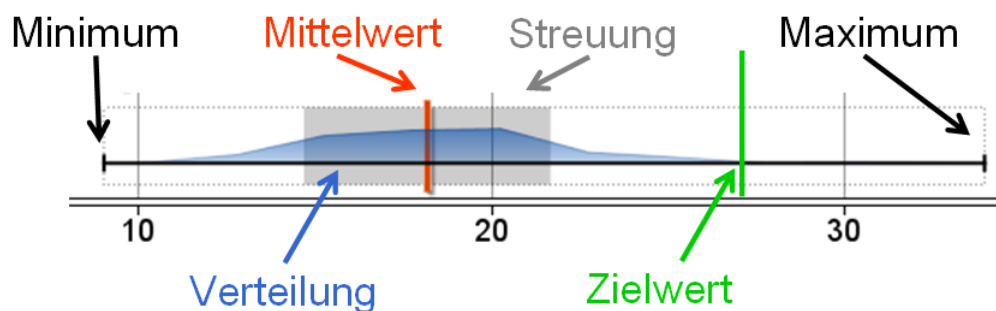


Abbildung B.1.: Erläuterung Verteilungsplot

Zentrales Element des Plots ist die Darstellung<sup>1</sup> der Verteilung der Werte, in der ablesbar ist, mit welchen Häufigkeiten die einzelnen Wertebereiche vertreten sind. Zur besseren Vergleichbarkeit werden zusätzlich statistische Werte (Minimum, Maximum, Mittelwert und Streuung) und der Zielwert eingezeichnet.

Eine Bewertung ist gut, wenn die Werte (repräsentiert durch die Verteilung und den Mittelwert) dicht und mit geringer Streuung beim Zielwert liegen.

<sup>1</sup>jeweils normiert auf das Intervall  $[0, 1]$



# C

## Anhang C.

---

# Parametrisierung der Optimierungsverfahren für den Vergleichstest

Für den Vergleich der Optimierungsverfahren wird eine Anzahl von 300 Funktionsauswertungen pro Optimierung vorgegeben. In den folgenden Tabellen sind die verwendeten Verfahrensparameter für Optimierungsverfahren aufgelistet.

### C.1. MCO

Parameter	Wert
Anzahl Punkte	300
Methode	gleichverteilt

### C.2. NSGA-II

Parameter	Wert
Populationsgröße	16
Generationen	36
Austausch pro Generation	50%
Kreuzungswahrscheinlichkeit	1
Mutationswahrscheinlichkeit	0.1-0.25
Kreuzungsindex	5
Mutationsindex	10

### C.3. Nelder-Mead

Parameter	Wert
max. Iterationen	90
PARETO-Scan	10
Zufällige Startpunkte	38
Reflektionsfaktor	1
Expansionsfaktor	2
Kontraktionsfaktor	0.5

### C.4. AHMOS

#### C.4.1. Globale Parameter

Parameter	Wert
Anzahl Funktionswertberechnungen	300
Anzahl Iterationen vor Hybridoperationen	5
Schranke für Nachiteration	0.25
Schranke für interne Ersatzmodellssuche	0.65
Generationenanzahl für Ersatzmodellssuche	100

#### C.4.2. Globale Suche mit NSGA-II

Parameter	Wert
alle Einstellungen wie C.2	

#### C.4.3. Lokale Suche mit Nelder-Mead

Parameter	Wert
max. Iterationen	15
Reflektionsfaktor	1
Expansionsfaktor	2
Kontraktionsfaktor	0.5

# Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, 20.Oktober 2011

.....

Steffen Kux

